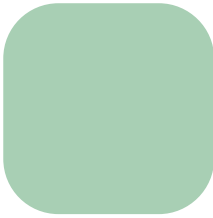


# virtual payment client integration guide



vpos canada



### **Copyright Statement**

Copyright © 2009 by Amex Bank of Canada. All rights reserved. No part of this document may be reproduced in any form or by any electronic or mechanical means, including information storage and retrieval systems, without the express prior written consent of Amex Bank of Canada.

Version 3.1.26.0  
05/20/2009

# contents

<b>preface</b>	<b>1</b>
audience .....	1
where to get help .....	1
<b>introduction</b>	<b>2</b>
about this document .....	2
related documents and materials .....	3
virtual payment client reference guide .....	3
merchant administration user guide .....	3
example code .....	3
terminology .....	4
<b>understanding e-payments</b>	<b>6</b>
what are e-payments? .....	6
the components of an e-payment solution .....	7
how e-payments transfer funds .....	7
about e-payment information flows .....	8
the merchant application .....	8
the virtual payment client .....	8
payment models .....	8
purchase model .....	8
authorisation/capture model .....	9
<b>preparing for integration</b>	<b>11</b>
integration models and communication methods .....	11
3-party payments integration model .....	11
2-party payments integration model .....	12
selection guidelines for integration models .....	13
when to use 3-party payments .....	13
when to use 2-party payments .....	13
when to combine 3-party and 2-party payments .....	14
prerequisites .....	15
support material and information .....	15
determine your integration model .....	15
determine the payment model .....	15
determine any advanced functionality .....	15
obtain an e-commerce merchant facility .....	16
provide your financial institution merchant number, terminal ID/s and MCC to your payment provider .....	16
look up your access code and secure hash secret in merchant administration .....	16
perform a basic test transaction using the supplied example code .....	17
determine the input and output fields .....	17
design and implement the integration .....	17
test your integration .....	18

conduct final pre-production testing.....	18
go live .....	18
commence live online payments .....	18
<b>virtual payment client integration guidelines</b>	<b>19</b>
reference fields.....	19
merchant transaction reference (vpc_MerchTxnRef) .....	19
virtual payment client order information (vpc_OrderInfo) .....	20
ensuring successful payments .....	20
manually check transaction results using merchant administration .....	20
automatically check the integrity of 3 party transactions using secure hash .....	21
<b>securing your payments</b>	<b>22</b>
protecting cardholder information using SSL.....	22
how do my cardholders know if my site is using SSL?.....	22
using 3-D Secure payment authentications .....	23
best practices to ensure transaction integrity.....	23
use a unique MerchTxnRef for each transaction attempt.....	23
check for a replay of a transaction .....	24
check that the field values in the response match those in the request.....	24
check for suspect transactions .....	24
use good password security for merchant administration .....	24
validate the SSL certificate of the payment server .....	24
additional features for 3-party transactions.....	25
<b>integrating 2-party payments</b>	<b>28</b>
2-party payments information flow .....	28
what the cardholder sees .....	29
<b>integrating 3-party payments</b>	<b>30</b>
3-party payment information flow.....	30
what the cardholder sees.....	31
integrating 3-party payments with virtual payment client.....	35
handle a transaction request.....	35
what the payment server does .....	36
handle a transaction response .....	36
handle session variables .....	37
additional 3 party functionality .....	38
3-party payments where the merchant collects the cardholder's card type .....	38
3-party payments where the merchant collects all the cardholder's card details .....	38
3-party payments using Verified by Visa™ and MasterCard SecureCode™ .....	39
<b>supplementary transactions</b>	<b>40</b>
address verification service (AVS).....	40
card holder name transactions .....	40
card security code (CSC/CVV2).....	41
card present transactions.....	42
external payment selection (EPS).....	42
merchant transaction source.....	43

merchant transaction frequency.....	43
referral message.....	44
referral transaction.....	44
airline ticket number.....	44
risk management.....	45
bank account type.....	47
payment authentication.....	47
payment authentication 3-D Secure transaction modes .....	48
mode 1 – implementing a 3 party authentication and payment transaction (payment server collects card details).....	53
mode 2 – implementing a 3 party authentication and payment transaction (merchant collects card details) .....	54
mode 3a – implementing a 3 party style authentication only transaction .....	55
mode 3b – implementing a 2-party style pre-authenticated payment transaction .....	57
<b>advanced merchant administration (AMA)</b> .....	<b>58</b>
capture.....	58
standalone capture.....	59
refund .....	59
standalone refund.....	59
void capture .....	59
void refund.....	60
void purchase .....	61
QueryDR.....	61
excessive captures .....	62
AMA shopping transaction history.....	62
AMA limited shopping transaction history.....	63
AMA financial transaction history .....	63
AMA spanned financial transaction history.....	64
<b>troubleshooting and FAQs</b> .....	<b>65</b>
troubleshooting.....	65
what do we do if a session timeout occurs? .....	65
what does a payment authentication status of "A" mean? .....	65
does the cardholder's internet browser need to support cookies? .....	65
what happens if a transaction response fails to come back? .....	66
frequently asked questions .....	68
what is an outage? .....	68
how do i know if a transaction has been approved? .....	68
can the payment server's payment pages be modified for a merchant?.....	68
is a shopping cart required? .....	68
what is merchant administration? .....	68
how much will it cost to keep the payment site running?.....	69
does the payment server handle large peaks in transaction volumes?.....	69
how long will an authorisation be valid on a cardholder account? .....	69
what is the RRN and how do I use it? .....	69
what is the difference between RRN, MerchTxnRef, OrderInfo, Authorizeld and TransNo? .....	70
advanced function compatibility.....	71
suggested merchant actions.....	72



# 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 Payment Client, please contact American Express Software Support toll free at 1-877-PEO-HOW2 (1-877-736-4692), from 9:00 A.M. to 7:30 P.M. Eastern Time, Monday through Friday.

# introduction

The 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 online store by using the functionality of the Virtual Payment Client.

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

## about this document

This document is the *Virtual Payment Client Integration Guide*. It is part of the Virtual Payment Client documentation set. It contains information on how to integrate Virtual Payment Client with the merchant's software.

Section	Description
<b>Understanding e-Payments</b> (see "Understanding e-Payments" on page 6)	Describes how e-Payments work.
<b>Preparing for Integration</b> (see "Preparing for Integration" on page 11)	Describes the various options and models you need to choose before commencing your integration.
<b>Securing your Payments</b> (see "Securing Your Payments" on page 22)	Describes the security features available for the Virtual Payment Client
<b>2-Party Payments</b> (see "Integrating 2-Party Payments" on page 28)	Describes the information flow and integration model for 2-Party Payments.
<b>3-Party Payments</b> (see "Integrating 3-Party Payments" on page 30)	Describes the information flow and integration model for 3-Party Payments.
<b>Supplementary Transactions</b> (see "Supplementary Transactions" on page 40)	Describes the supplementary fields available on the Payment Server, and the additional data that you must add to the Transaction Request if you want to implement the optional functionality.



Section	Description
<b><i>Advanced Merchant Administration (AMA) Transactions</i></b> (see "Advanced Merchant Administration (AMA)" on page 58)	Describes the flows for each AMA transaction/query, and the data requirements for each stage of the transaction/query.
<b><i>Troubleshooting and FAQs</i></b> (see "Troubleshooting and FAQs" on page 65)	Describes suggestions and solutions to problems that may occur with your integration, and answers to commonly asked questions.

## related documents and materials

The following provide additional information that may be useful to you.

### virtual payment client reference guide

This *Virtual Payment Client Integration Guide* is designed to be used with the *Virtual Payment Client Reference Guide*, which details the various types of transactions of the Virtual Payment Client's API methods, plus its inputs and outputs.

### 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 to illustrate the use of the Virtual Payment Client API.

## terminology

Term	Description
<b>Access Code</b>	<p>The access code is an identifier that is used to authenticate you as the merchant while you are using the Virtual Payment Client.</p> <p>The access code is generated and allocated to you by Merchant Administrator.</p>
<b>Acquirer Bank</b>	<p>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.</p>
<b>Bank</b>	<p>The bank with which you have a merchant facility that allows you to accept online credit card payments.</p>
<b>Capture</b>	<p>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.</p>
<b>Financial Institution (FI)</b>	<p>See Bank.</p>
<b>Issuing Bank</b>	<p>The financial institution that issues credit cards to customers.</p>
<b>Merchant Administration</b>	<p>Merchant Administration allows you to monitor and manage your electronic transactions through a series of easy to use, secure web pages.</p>
<b>Payment Provider</b>	<p>The Payment Provider acts as a gateway between your application or website and the financial institution.</p> <p>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.</p> <p>Your Payment Provider may be your acquirer bank or a third party technology services provider.</p>
<b>Payment Server</b>	<p>The Payment Server facilitates the processing of secure payments in real-time over the Internet between your application/website and the Payment Provider.</p> <p>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.</p>
<b>Purchase</b>	<p>Purchase is a single transaction that immediately debits the funds from a cardholder's credit card account.</p>

Term	Description
<b>RRN</b>	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.
<b>Transaction Request</b>	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.
<b>Transaction Response</b>	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.
<b>Virtual Payment Client</b>	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.
<b>Transaction</b>	A combination of a Transaction Request and a Transaction Response. For each customer purchase or order, merchants may issue several transactions.

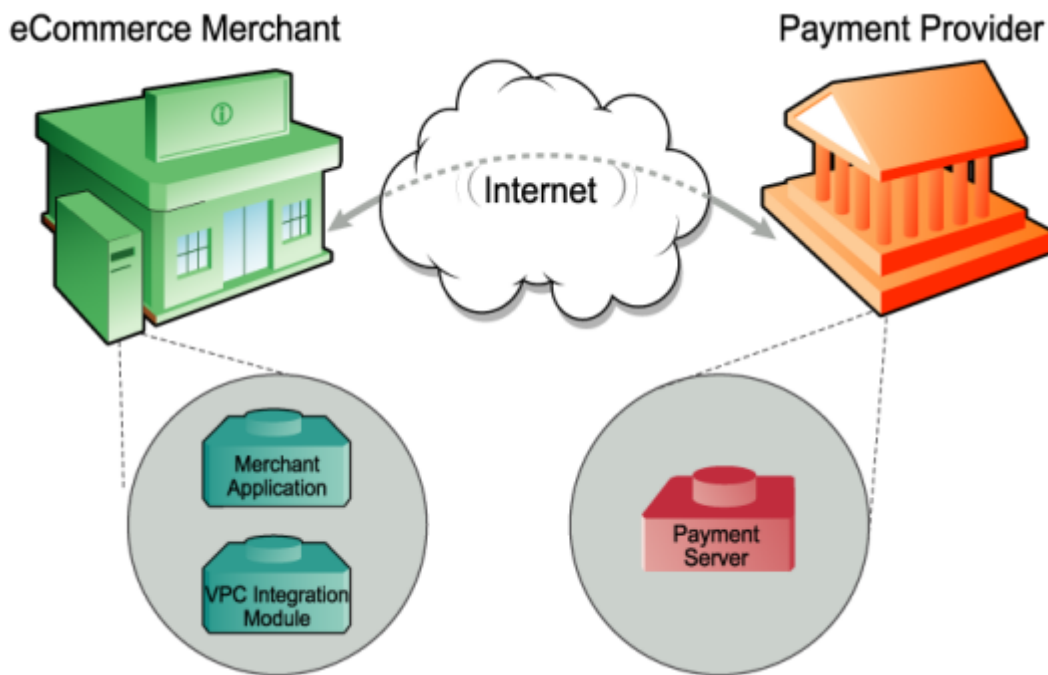
# understanding e-payments

This section is an overview of electronic payments or e-Payments.

## what are e-payments?

e-Payments are secure real time payments that transfer funds (using the Internet) between a cardholder and the merchant's financial institutions. e-Payments require secure communication between all components of the e-Payment process.

e-Payments are represented in the following diagram:



## the components of an e-payment solution

An end-to-end e-Payment solution is made up of the following components:

- **The Merchant application** is a business application/website on the merchant's system that uses Virtual Payment Client to process payments.
- **The Integration module** is a communication bridge between the merchant application and Virtual Payment Client.
- **Virtual Payment Client** provides secure communication between the merchant application and the Payment Server. Virtual Payment Client can be integrated with a number of systems including merchant applications, Interactive Voice Response (IVR) systems, and integrated ERPs.
- **Payment Server** processes merchant Transaction Requests.
- **The Payment Provider** enables the merchant to accept payments online.

## how e-payments transfer funds

e-Payments transfer funds using the following steps:

- 1 The cardholder purchases goods/services from the merchant (for example, in person, using the Internet, or over the phone).
- 2 The merchant application sends a Virtual Payment Client Transaction Request (via the Payment Server) to the merchant's Payment Provider.
- 3 The merchant's Payment Provider directs the request to the cardholder's bank.
- 4 The cardholder's bank debits the cardholder's account and transfers the funds to the merchant's account at the merchant's Payment Provider.

## about e-payment information flows

This section describes how information is transferred between the merchant application and the Payment Server.

### the merchant application

To process a payment, the merchant application must send the required information to the Payment Server. The merchant application must create a message in a specified format to send this information using the Virtual Payment Client, which is part of the Payment Server using two messages:

- **Transaction Request** is sent to the Virtual Payment Client in the Payment Server to provide transaction information.
- **Transaction Response** is returned from the Payment Server using the Virtual Payment Client to indicate the outcome of the transaction (that is, successful or otherwise).
- A **Transaction** is the combination of a Transaction Request and a Transaction Response. For each customer order, merchants may issue several transactions.

### the virtual payment client

- Receives the Transaction Request from the merchant application
- Sends the information to the Payment Server
- Receives the result from the Payment Server, creates a response in the appropriate format and forwards it to the Merchant Application.

## payment models

Virtual Payment Client supports the most commonly used payment models in the e-Payments process. These include the Authorisation/Capture model.

Payment Integration models are described in *Preparing for Integration* (see "Preparing for Integration" on page 11).

### purchase model

Purchase is the most common type of payment model used by merchants to accept payments. A single transaction is used to authorise the payment and initiate the debiting of funds from a cardholder's credit card account.

This is typically used when the goods will be delivered immediately following a successful transaction.

## authorisation/capture model

The authorisation/capture payment type is a two step process. The merchant uses an Authorisation transaction to reserve the funds.

### authorisation in the auth/capture model

The Authorisation (Auth) transaction verifies that the card details are correct and may/may not also reserve the funds, depending on the merchant's Payment Provider. To find out what models are available to you, contact your Payment Provider.

The authorisation is used to ensure that the cardholder has sufficient funds available against their line of credit. The full amount of the order is sent to the card Issuing Bank to verify the details against the cardholder's card account. The authorisation does not debit funds from the cardholders account, but reserves the total amount, ready for the capture transaction to debit the card and transfer the funds to your account.

The cardholder's credit limit is reduced by the authorised amount. If they make another transaction, this current authorisation transaction is taken into account and comes off the cardholder's available funds as though the transaction had already taken place. This authorisation reserves the funds for a predetermined period of time, (such as 5-8 days), as determined by the card scheme and the cardholder's card issuing rules.

The API does not have a method to void an Authorisation transaction so it must fade out at the end of the appropriate period. Authorisation transactions do not appear in the cardholder's account records, only the capture transactions appear.

The Authorisation transaction uses the same API as the standard payment transaction used in the Purchase model where a Capture transaction is not required. The only difference is how the merchant profile is configured with the Payment Provider.

### **Pre-Authorisation/Purchase Mode**

This is a variation of the Authorisation/Capture process where your Payment Provider verifies the card details with the card issuing institution, and if the transaction were carried out at this exact point in time whether the transaction would be successful. No funds are reserved on the cardholder's account.

If the cardholder performed another transaction between the pre-authorisation transaction and the purchase transaction that used up all the available funds on the card, then the later purchase transaction may fail due to lack of funds (if applicable). The merchant must include the full amount in their Pre-authorisation transaction as the Payment Server uses it to ensure that later Purchase transactions do not exceed the total amount specified in the Pre-authorisation transaction.

The Pre-Authorisation and Purchase transactions in this mode use exactly the same API as the Authorisation/Capture transactions outlined earlier. The only difference is how the merchant's Payment Provider actions the two transactions.

### **Nominal Auth/Purchase Mode**

This is a variation of the Pre-authorisation/Purchase model where the Payment Server strips the value in the Authorisation transaction and substitutes a nominal transaction value. The acquiring bank checks the card details with the issuing card institution to ensure they are correct. No funds at all are reserved on the card. The merchant must include the full amount in their Nominal Authorisation transaction as the Payment Server uses it to ensure that later Purchase transactions do not exceed the total amount specified in the Nominal Authorisation transaction.

The Nominal Auth/Purchase transactions in this mode use exactly the same API as the Authorisation/Capture transactions outlined earlier. The only difference is how the merchant's Payment Provider actions the two transactions.

### **capture in the auth/capture model**

The capture transaction refers back to the initial authorisation transaction, and transfers the funds from a cardholder's card into the merchant's account.

The merchant can perform any number of capture transactions on the original Authorisation transaction; however, the total of all the amounts from all the captures cannot exceed the original authorised amount. For example, the merchant may not have the full ordered amount of goods in stock. Hence they ship what they do have and capture the funds from the cardholder accordingly. Later when the remaining goods are shipped the merchant performs another capture transaction that refers back to the same initial authorisation transaction. This causes the remaining funds to be transferred from the cardholder's account to the merchant's account. The capture transactions will be successful, provided:

- The total amount for the all captures do not exceed the original Authorisation amount, and
- The card issuing institution has not expired the original Authorisation transaction.



## preparing for integration

Before you start integrating, you must determine if your Payment Provider supports the functions that you require. This will determine the transaction types you can or cannot integrate.

### integration models and communication methods

There are two ways that you can communicate with the Payment Server to process transactions, the Redirect method and the Direct method. The method you choose is directly related to the Integration Model, either 3-Party or 2-Party, that you use. You may use both methods concurrently if necessary, for example, you may have a Web Store that uses 3-Party, and at the same time a Call Centre taking phone orders using 2-Party. Both applications could be using the Payment Client at exactly the same time.

### 3-party payments integration model

3-Party Payments allow the Payment Server to manage the payment pages and collect the cardholder's card details on your behalf. The Payment Server's payment pages could be Bank or Payment Provider branded to help assure the cardholder of a secure transaction. The advantage of 3-Party payments is that the complexity of securely collecting and processing card details is handled by the Payment Server, allowing you to focus on your application's part of the payment process.

However, 3-Party Payments do also allow you to collect card details on your web site and pass them through with the other transactional details. If this is done the Payment Server does not display any 3rd party branded pages, keeping the branding consistent throughout the whole transaction, except the 3-D Secure pages if the merchant and the cardholder are both enrolled in this antifraud initiative. To do this you would have to comply with the same obligations associated with 2-Party payments.

The 3-Party Redirect method only works for web applications where a web browser is involved. This method is also required to implement 3-D Secure antifraud initiatives of Verified by Visa™ and MasterCard SecureCode™. The redirect method works with most network configurations and you do not need to take into account proxy servers as the information is communicated to and from the Payment Server using the cardholder's Internet browser.

The cardholder's Internet browser is redirected to take the Transaction Request to the Payment Server to process the transaction. After processing the transaction, the cardholder's Internet browser is returned to a web page that you nominate in the transaction together with a Transaction Response. The Transaction Response processing of the receipt information finalises the transaction.

The 3 parties involved in a 3-Party transaction are the merchant, the payment provider and the cardholder. The cardholder's browser provides the redirect method to communicate the information between the merchant and the payment provider. This is an asynchronous connection and the cardholder leaves your web site to go to the Payment Server, which means the transaction is broken or disrupted into 2 distinct sessions, the creation of the Transaction Request and the processing of the Transaction Response.

Because of this, you may be required to capture session variables and include them in the Transaction Request so they can be passed back appended to the Transaction Response for restoring the original web session.

## 2-party payments integration model

Merchants who want full control over the transaction and want to manage their own payment pages use the 2-Party integration model. Implementing 2-Party requires you to securely collect the cardholder's card details and then use the Virtual Payment Client to send the Transaction Requests directly to the Payment Server. This is also called the merchant-managed, or direct model. This model means that you are responsible for securing the cardholders card number and details.

The 2-Party does not allow you to implement the 3-D Secure anti-fraud initiatives of Verified by Visa™ and MasterCard SecureCode™.

The 2 parties involved in a 2-Party transaction are the merchant and the payment provider. The merchant communicates directly through the Virtual Payment Client to the Payment Server and back again. This is a synchronous connection and the cardholder does not leave your site, which means the session is not broken or disrupted.

The Direct method is also used for advanced Payment Server operations such as captures, refunds, voids and queries. Your application communicates to the Payment Server via the Virtual Payment Client, so you need to take into account working with proxy servers.

The methods used to work with these proxy servers will vary slightly depending on the programming language used by your application.

## selection guidelines for integration models

Use the following guidelines to select an integration model, depending on your application, preferred communication method, security needs, and future plans.

### when to use 3-party payments

Consider using 3-Party Payments if:

- You are integrating a web browser-based application only. Call centres, IVRs and other applications cannot use this transaction mode.
- You want to, either now or in the future, increase security by using 3-D Secure authentication (for example, Verified by Visa™ and MasterCard SecureCode™).
- It is acceptable to have the cardholder's browser redirected away from your web site to the Payment Server.
- You want the Payment Provider to collect and manage the cardholder's card details and to manage the associated security and privacy issues.
- It is acceptable to display Payment Provider-branded pages in the payment flow.

**Note:** If you require branding to be consistent throughout a 3-Party transaction, you can collect card details and include them into the Transaction Request. However the higher risk and responsibility of collecting card details remains the same as in a 2-Party transaction.

### when to use 2-party payments

Consider using 2-Party Payments if:

- You are willing to collect card details and manage the associated security and privacy issues (VISA AIS, MasterCard SDP and so forth).
- You are integrating an application with the Virtual Payment Client (for example, web, call centre, billing application, Interactive Voice Response (IVR) system) that does not use 3-D Secure authentication (for example, Verified by Visa™ and MasterCard SecureCode™). For more information see **Payment Authentication** (see "Payment Authentication" on page 47).
- You do not want the cardholder's browser to be redirected away from your web site to the Payment Server for payment processing.
- You do not want to display Payment Provider-branded pages in the payment flow.

## when to combine 3-party and 2-party payments

Consider using both 2-Party and 3-Party if any of the following are true:

- You want to use a combination of 3-Party for Web and 2-Party for call centre/IVR/other applications.
- You have a web application in which you want to perform some form of repeat payment, as in a subscription, where you want to take advantage of 3-D Secure authentication for the first payment and then use 2-Party payment transactions for each subsequent installment payment. (You must capture and store the card details to do this.)
- You are willing to use 3-Party transactions for payments and are also using other transactions like refunds and queries, which are all 2-Party mode transactions.

**Note 1:** If you are collecting card details and want to implement 3-D Secure authentication, you only need to perform 3-Party transactions for those transactions that require 3-D Secure authentication like MasterCard and Visa. Other transactions that don't use 3-D Secure authentication such as Bankcard and American Express can be performed using 2-Party transactions as they don't support Authentication.

**Note 2:** Advanced Merchant Administration functions such as captures, refunds, voids and queries all use the 2-Party style of transaction, so if you need to use any of these transaction types through the Virtual Payment Client, you will also need to install the Virtual Payment Client with the 2-Party options installed. These operations, captures, refunds, voids and queries carry no higher risk than 3-Party as you do not need to pass in cardholder card information to carry out these transaction types.

## prerequisites

This section lists the requirements and basic steps you need to take to build a successful integration.

## support material and information

You must have the following:

- Virtual Payment Client Reference Guide
- Example Code for your site (written in ASP, JSP, PHP and Perl)
- Test Card setup document

## determine your integration model

You must choose either:

- 3-Party Payments Integration Model
- 2-Party Payments Integration Model
- Combination of 2-Party and 3-Party Integration

For more information, please refer to **Integration Models** (see "Integration Models and Communication Methods" on page 11).

## determine the payment model

- **Purchase** – requires a single transaction to transfer funds from the cardholder's account to your account.
- **Authorisation/Capture** – requires two transactions, the Authorisation, followed separately by a Capture. For more information, see **Authorisation/Capture** (see "Authorisation/Capture Model" on page 9).

## determine any advanced functionality

The available advanced functionality includes:

- Verified by Visa™ and MasterCard SecureCode™. See **Securing your Payments** (see "Securing Your Payments" on page 22).
- Capture.
- Refund.
- Voids.
- QueryDR.

## obtain an e-commerce merchant facility

After your E-Commerce merchant facility has been approved, your Financial Institution (FI/Bank) will provide the following information to you:

- **Merchant Number**  
Without this information you cannot perform any transactions. It ensures your settlement funds from successful payments are deposited to your correct account.
- **Terminal ID/s**  
The Payment Provider's identifier/s for the terminal/s used to process payments.
- **Merchant Category Code (MCC)**  
A four-digit code allocated to you by the Payment Provider based on your business type.

## provide your financial institution merchant number, terminal ID/s and MCC to your payment provider

This information is needed to establish your merchant profile with your Payment Provider. Your merchant profile holds your configuration data including Financial Institution account details and your access credentials to the payments service.

Your Payment Provider will then issue you with a unique Payment Server Merchant ID identifying you to the Payment Server and also provide you with a User Name and Password for accessing Merchant Administration to manage your transactions.

## look up your access code and secure hash secret in merchant administration

You need your Virtual Payment Client Access Code and Secure Hash Secret before starting your integration:

- **Access Code**  
The access code uniquely authenticates a merchant and their Merchant ID on the Payment Server.
- **Secure Hash Secret (3-Party) only**  
The Secure Hash Secret is a key used as the initial piece of encryption data to create an MD5 Secure Hash. This ensures transaction data is not tampered with while in transit to the Virtual Payment Client.

Your access code and secure hash secret can be found in Merchant Administration in the Setup menu option on the Configuration Details page. Please refer to your *Merchant Administration User Guide* for details on how to locate your Access Code and Secure Hash Secret.

## perform a basic test transaction using the supplied example code

Successful completion of a transaction using the standard example code before you implement the integration with your application:

- Validates that your system is setup correctly
- Ensures basic functionality is available

The standard example code covers common web server scripting languages. You must select the appropriate example for your specific web environment.

**Note:** The standard example code contains examples of how to integrate your application and may not fully correspond with the feature set that you have chosen to implement.

## determine the input and output fields

Determine how you are going to get the Transaction Request input fields and where to store the Transaction Response output fields in your application.

You need to consider:

- **Session Variables** – When using 3-Party payment (with or without card details) some applications may require session variables to be collected and sent to the Payment Server in the Transaction Request. The session variables are returned in the Transaction Response allowing your application to continue with the order process using the same application session. For more information on session variables see **Session Variables** (see "Handle Session Variables" on page 37).

Session variables are not required when using the Direct or 2-Party communication method as the session is not broken while performing a transaction.

- **Merchant Transaction Reference (vpc\_MerchTxnRef)** – You need to determine how you are going to produce a unique value for a transaction using the vpc\_MerchTxnRef field. For more, see Merchant Transaction Reference.

## design and implement the integration

You are now ready to payment enable your application. This step requires a web developer familiar with both your application and the web programming language used in your web environment.

This guide provides the information and best practice guidelines to assist you with this task. You should also refer to the example code and *Virtual Payment Client Reference Guide* for further assistance.

## test your integration

You need to test your integration by performing test transactions. The Payment Server has a test acquirer facility to test all the different response codes that you are likely to encounter in a live environment.

Performing test transactions allows you to test your integration, so that you won't encounter problems when processing real transactions. For more information, please refer to the Test Card set up document supplied to you by your Payment Provider, located in the *Payment Client Reference Guide* in "Appendix C – Test Transport Response".

## conduct final pre-production testing

It is recommended that you follow standard IT practices and complete final pre-production testing with live credit cards to validate that end-to-end functionality works correctly, including successful settlement of funds from your financial institution.

Remember you can always refund these test transactions.

## go live

Once you are satisfied that your integration works correctly, please advise your Payment Provider that your testing has been successfully completed.

Your Payment Provider will validate your testing results and then provide you with your production profile and instructions on how to change your website from test mode to live production mode, allowing you to process live transactions with your Financial Institution (bank).

## commence live online payments

You should now be ready to launch your payment enabled application and start processing online payments from your cardholders.



# virtual payment client integration guidelines

This section describes certain key issues that you must take into account while writing your integration code.

## reference fields

It is helpful to have an understanding of the following fields when integrating your payment application.

### merchant transaction reference (**vpc\_MerchTxnRef**)

The **vpc\_MerchTxnRef** field is a unique identifier that the merchant assigns to each transaction. This unique value is used by the merchant to query the Payment Server database to retrieve a copy of a lost/missing transaction receipt using a 2-Party QueryDR function. This is the only purpose of **vpc\_MerchTxnRef**. If you are not going to use the QueryDR function then you don't need to be concerned about the **vpc\_MerchTxnRef** field. Simply copy the **vpc\_OrderInfo** field value into the **vpc\_MerchTxnRef** field.

You can use a value like an order number or an invoice number as the foundation for the **vpc\_MerchTxnRef**. However, if you want to allow cardholders to repeat a transaction that was declined and you want to keep the same order number (or invoice number), you must modify the **vpc\_MerchTxnRef** for each subsequent attempt, by appending extra characters for each attempt. For example **vpc\_MerchTxnRef** = '1234/1' on first attempt, '1234/2' on second attempt, and '1234/3' on third attempt, etc.

Under a fault condition, such as if the Transaction Response does not arrive back at the merchant's site due to a communication error, you may need to check if the transaction was carried out successfully. A unique **vpc\_MerchTxnRef** makes cross-referencing the transactional data easier.

This is achieved by performing a QueryDR command that will search the Payment Server's database for the transaction, based on the **vpc\_MerchTxnRef**. If you have not given each transaction attempt a unique **vpc\_MerchTxnRef** number, then there will be multiple results and the QueryDR command may not return the correct transaction attempt you are looking for as it only returns the most recent transaction information.

## virtual payment client order information (vpc\_OrderInfo)

**vpc\_OrderInfo** is an identifier provided by you to identify the transaction on the Payment Server database. This value will be displayed in the Merchant Administration portal when manually searching transactions. It can be an order number, an invoice number, or a shopping cart number. The **vpc\_OrderInfo** field can remain static for each transaction but you should have a unique **vpc\_MerchTxnRef** for each transaction as outlined above for use with the QueryDR function.

The **vpc\_OrderInfo** field is used to send a merchant specified reference, for example, Merchant's Invoice No = **vpc\_OrderInfo** and can be used to search for another in Merchant Administration.

## ensuring successful payments

It is recommended that you consult with security experts with experience in your web environment to ensure that your security implementation is suitable for your needs.

An issue that merchants have to deal with when implementing payments solutions is "How to be sure that you get paid for the goods you ship?"

To ensure that you will be paid means that you need to ensure the response integrity and identification/authentication of the Payment Server during the payment process.

To ensure that you will be paid you can:

- Where possible, implement the 3-Domain Secure services of Verified by Visa™ and MasterCard SecureCode™. See **Securing your Payments** (see "Securing Your Payments" on page 22).
- Manually check all transaction results at the Payment Server by logging into Merchant Administration before fulfilling each order.
- Automatically check transaction results at the Payment Server before fulfilling each order by using the Query DR functionality (if available).
- Automatically check and verify the integrity of each message when the payment is performed by using the Secure Hash functionality.

## manually check transaction results using merchant administration

This process suits merchants with very low volume sales. It requires you to log in to your Merchant Administration and run a report to view the OrderIDs and then match them against the orders logged on your website. If they match, you can ship the product, and follow up on, or discard orders where the payment failed, or the payment does not exist.

The risk is the possible human error of matching OrderIDs with the cardholder's orders manually. Also as volumes grow, the risk may become significant as would the time and cost involved completing the task.

## automatically check the integrity of 3 party transactions using secure hash

The Secure Hash is used to detect the cardholder modifying a Transaction Request or Transaction Response when passing it through their cardholder's browser. Using the Secure Hash ensures a high level of trust in the transaction result.

The benefit of using Secure Hash is that the integrity of each response can be checked without having to create a new SSL connection to the Payment Server for each transaction.

The Secure Hash Secret must be kept secret to provide security and should be changed periodically for this method to be effective.

The Secure Hash method is **only applicable when using the 3-Party** Payments integration model.

## securing your payments

This section describes the security features available for the Virtual Payment Client. It is recommended that you understand this section before you start integrating your application with the Virtual Payment Client.

### protecting cardholder information using SSL

All websites collecting sensitive or confidential information need to protect the data passed between the cardholder's Internet browser, the application and the Payment Server.

SSL is a security technology that is used to secure web server to Internet browser transactions. This includes the securing of any information (such as a cardholder's credit card number) passed by an Internet browser to a web server (such as your web 'Shop & Buy' application). SSL protects data submitted over the Internet from being intercepted and viewed by unintended recipients.

The Payment Server is responsible for securing the cardholder details when you implement the 3-Party Integration Model. It uses SSL, which encrypts sensitive financial data to provide a secure transmission between a cardholder and the Payment Server.

When implementing the 2-Party or 3-Party Integration models you must ensure your application presents a secure form using SSL. You should also consider using a secure form in your application when collecting confidential information such as cardholder addresses.

**Note:** SSL should also be used for 3-Party. This avoids the possible browser alert message indicating that the cardholder is being redirected to an unsecured site. This can happen when the cardholder's browser is being redirected back to the merchant's web site with the encrypted Transaction Response for decryption.

If the cardholder clicks on 'No' within the pop-up message, then neither the cardholder nor the merchant will receive any receipt details.

### how do my cardholders know if my site is using SSL?

When a cardholder connects to your application using SSL they will see that the http:// changes to https:// and also a small gold padlock will appear in their Internet browser, for example:



Whenever an Internet browser connects to a web server (website) over https:// – this signifies that the communication with the Payment Server will be encrypted and secure.

You can alert your customers to this fact so they know what to look for when transacting on your web site.

## using 3-D Secure payment authentications

3-D Secure Payment Authentication contains Verified by Visa™ and MasterCard SecureCode™, which are designed to minimise credit card fraud, by attempting to authenticate cardholders when performing transactions over the Internet. Authentication ensures that a legitimate owner is using the card as the Payment Server redirects the cardholder to their card issuing institution where they must enter a password that they had previously registered with their card issuer.

To use Verified by Visa™ and MasterCard SecureCode™, you need to request a 3-D Secure enabled merchant profile from your Payment Provider and implement the 3-Party Payment Integration Model.

**Note:** Payment authentication is only supported for web transactions using 3-Party Payments through a browser. This is because the cardholder's web browser must be redirected to their card issuing bank.

For the information flow of a 3-Party Authentication & Payment transaction please see **Authentication Information Flow** (see "Information Flow of a 3D-Secure Authentication/Payment transaction" on page 50).

## best practices to ensure transaction integrity

The following Best Practices are guidelines only. It is recommended that you consult with security experts with experience in your web environment to ensure that your security is appropriate for your needs.

### use a unique MerchTxRef for each transaction attempt

Each transaction attempt should be assigned a unique transaction reference ID. Most applications and web programming environments will generate a unique session for each cardholder, which can be used as the unique merchant transaction reference ID. You can alternatively create a unique reference ID by combining an order/invoice number with a payments attempt counter. You may also consider appending a timestamp to the transaction reference ID to help ensure that each one is unique.

Before sending a transaction to the Payment Server, you should store this unique transaction reference ID with the order details in your database. The merchant transaction reference ID is returned in the Transaction Response.

The unique transaction reference ID is required for you to reliably use the QueryDR function to retrieve the transaction details you may be searching for. For example, if a transaction is reported as lost or missing, you can use QueryDR to locate it.

## check for a replay of a transaction

You should check each Transaction Response to ensure that your unique Merchant Transaction Reference ID (**vpc\_MerchTxnRef**) matches that order, and that it does not correspond with any previous order that has already been processed.

## check that the field values in the response match those in the request

You should ensure that important fields such as the amount and the merchant transaction reference ID in the Transaction Response match up with the values input to the original Transaction Request.

## check for suspect transactions

Common things to look out for are:

- Use of free/anonymous E-mail by the cardholder
- Different Ship To and Bill To addresses
- Foreign orders or shipments from countries with reputations for high fraud activities
- High-priced orders
- Multiples of the same item

**Note:** It is recommended that you do not store any credit card information in your web site database. If you must store credit card numbers, they should be securely hardware encrypted, or you should store them as masked values (for example, 498765XXXXXX769).

## use good password security for merchant administration

It is highly recommended that you choose a password that is difficult to guess and change your password regularly. A good password should be at least 8 characters and should contain a mix of capitals, numbers and special characters.

## validate the SSL certificate of the payment server

It is highly recommended that you validate the SSL certificate of the Payment Server whenever you connect to the Payment Server. The Payment Server SSL certificate is issued by an industry standard Certificate Authority such as Verisign or Thawte whose root certificate should already be available in your web environment.

**Note:** Please consult a web developer if you are not familiar with validating SSL certificates or exporting certificates from websites.

Always ensure the server is a trusted source.

## additional features for 3-party transactions

The following features apply to 3-Party transactions only.

### detect alteration of requests and responses using secure hash

The Virtual Payment Client uses MD5 Signature Hashes, which play a role in transaction security as they are used to detect whether the Transaction Request and Transaction Response has been tampered with. The Secure Hash Secret is generated by the Payment Server and assigned to you. It is a unique value for each merchant and made up of alphanumeric characters. Only you and the Payment Server know what the Secure Hash Secret value is.

Your Secure Hash Secret is used along with the Transaction Request details to generate an MD5 Secure Hash, which is appended to the Transaction Request. Because the Payment Server is the only other entity apart from your application that knows your Secure Hash Secret, it is the only other entity that can recreate the same Secure Hash.

- If the Secure Hash recreated by the Payment Server is equal to the Secure Hash sent in the Transaction Request, it means that the Transaction Request has not been tampered with. The Payment Server will continue to process the payment.
- If the Secure Hash recreated by the Payment Server does not equal the Secure Hash sent in the Transaction Request it can be assumed the data has changed in transit. The Payment Server will not process the payment and return the cardholder to your site with an error message in the Transaction Response.

After processing the transaction, the Payment Server uses your Secure Hash Secret and the Transaction Response details to generate a Secure Hash which is sent to your application. Your application uses the Secure Hash Secret and the Transaction Response details received from the Payment Server to also generate a Secure Hash. If your generated Secure Hash matches the Secure Hash sent by the Payment Server the Transaction Response has not been tampered with.

If your generated Secure Hash does not match the Secure Hash sent by the Payment Server the Transaction Response has been tampered with and should be checked against the data stored in the Payment Server. This can be done by using an automatically QueryDR (if available) or manually using Merchant Administration.

The Secure Hash Secret is never sent from the application to the Payment Server or from the Payment Server to the application. It is held securely at both sites and is only used as a seed in the generation of the Secure Hash.

The use of the Secure Hash Secret is strongly recommended despite the possibility that the Secure Hash can be made optional if you have the **mayOmitHash** privilege set in your Merchant profile on the Payment Server. To enable this privilege, you need to contact your Payment Provider.

For more information on Secure Hash Secret, see **Store Secure Hash Secret securely** (see "Store Secure Hash Secret Securely" on page 27).

## creating an MD5 signature for 3-party transactions

The merchant code creates the MD5 Secure Hash value on the Transaction Request data. The Virtual Payment Client creates another MD5 Secure Hash value and sends it back to the merchant in the Transaction Response.

The Secure Hash is a Hex encoded MD5 output of a concatenation of all the data parameters. The order that the data parameters are hashed in is extremely important as different transactions contain different data fields so rather than giving the explicit order for each parameter, the order that parameters are hashed in should follow the following rules:

- The Secure Hash Secret data is always first.
- Then all parameters are concatenated to the secret in alphabetical order of the parameter name. More specifically, the data sort should be in ascending order of the ASCII value of each parameter's name, 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**'.
- Fields must not have any separators between them and must not include any null terminating characters or the like.

For example, if the secret is **0F5DD14AE2E38C7EBD8814D29CF6F6F0**, and the Transaction Request includes only the following parameters:

Field Name	Example Value
vpc_MerchantId	MER123
vpc_OrderInfo	Order456
vpc_Amount	2995

In ascending alphabetical order, the input to the MD5 Secure Hash creation routine would be:

```
0F5DD14AE2E38C7EBD8814D29CF6F6F02995MER123Order456
```

This string is then Hex encoded and then passed through the merchant's MD5 Secure Hash generator in the programming language the merchant is using. This output (for example, a value of **68798ab0259eb01be7bbe2a807171f83**) is then included in the Transaction Request using the vpc\_SecureHash field.

The Virtual Payment Client also includes the vpc\_SecureHash in the Transaction Response so the merchant can check the security of the receipt data. This is performed by first stripping off the vpc\_SecureHash, and then performing the same steps as creating an MD5 Secure Hash for the Transaction Request, but using the received Transaction Response data fields instead. The received vpc\_SecureHash is then compared with the MD5 Secure Hash calculated from the Transaction Response data.

If both MD5 signatures are the same, the data has not been changed in transit. If they are different the data needs to be doubled checked, perhaps using a QueryDR command.



To assist merchants, the Virtual Payment Client always returns the Transaction Response results with the Secure Hash last, and all other parameters in ascending alphabetical order.

### **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*.

## integrating 2-party payments

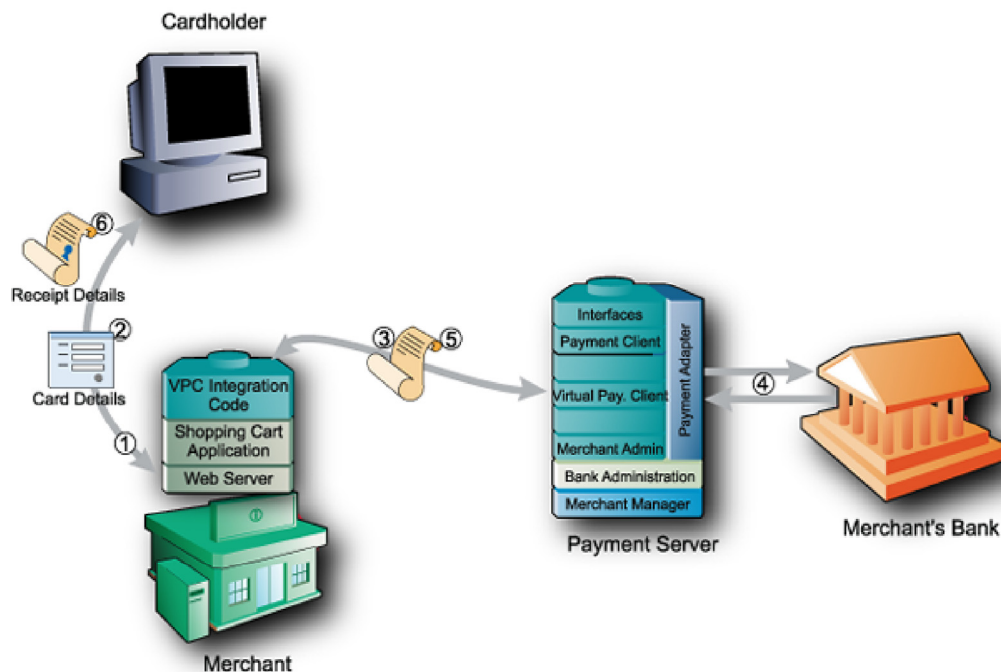
In the 2-Party Integration Model, a cardholder places an order and provides their card details (card type, card number and expiry date) to you by **Mail Order** or by **Telephone Order (MOTO transactions)** including Interactive Voice Response (IVR) systems, or some card present application like a ticketing system.

You can implement the 2-Party Integration Model if you prefer cardholders to provide their card details (card type, card number and expiry date) to you rather than to the Payment Server.

2-Party Payments carry a higher risk than 3-Party, as you are responsible for protecting the cardholder's card details.

### 2-party payments information flow

The following is the information flow in a 2-Party transaction.



- 1 A cardholder decides to make a purchase and provides their card details directly to your online store.
- 2 Your application collects the details of the cardholder's order.
- 3 In addition it formulates the Transaction Request and sends it using a HTTPS POST over the Internet to the Payment Server via the Virtual Payment Client.
- 4 The Payment Server passes the transaction to the merchant's acquirer bank for processing.

- 5 After processing, the Payment Server generates a Transaction Response and passes it via the Virtual Payment Client to your online store. The Transaction Response shows whether the transaction was successful. The results can be stored by you for future reference.
- 6 A receipt is generated and either immediately passed to the cardholder or included when shipping the goods.

## what the cardholder sees

In 2-Party Payments over the Internet the cardholder is presented with two pages:

- 1 The merchant's application checkout page.
- 2 The merchant's application receipt page.

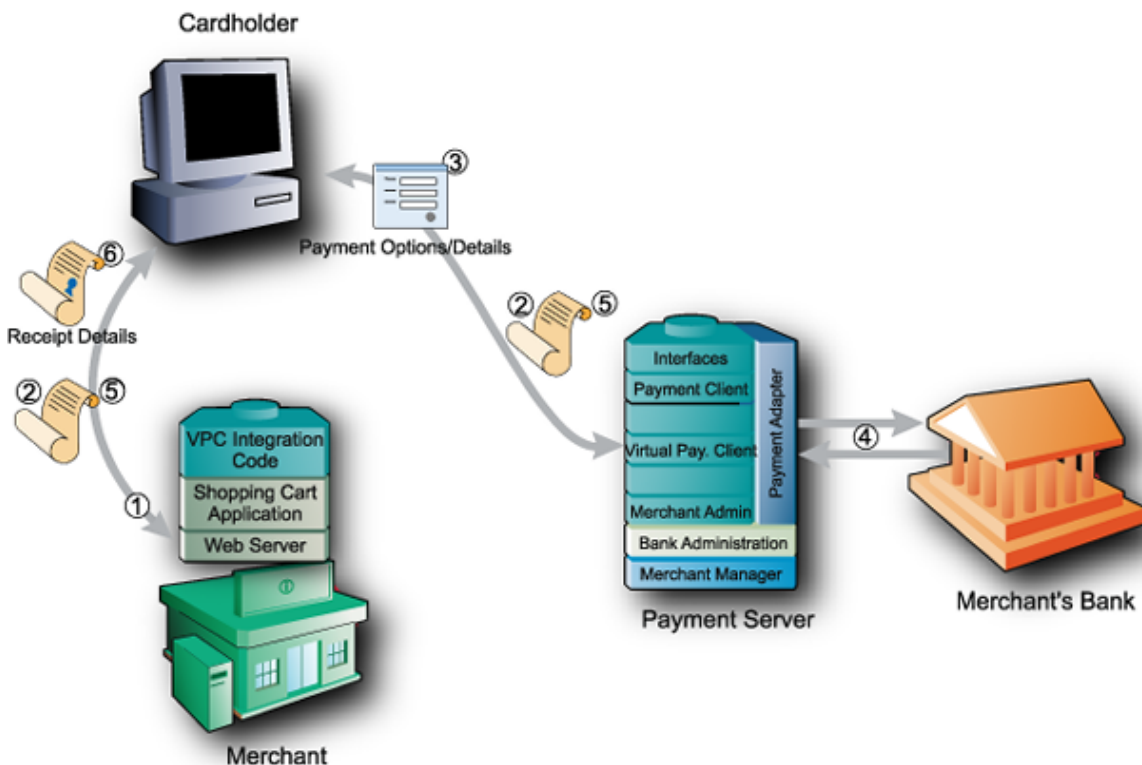
**Note:** Although you can implement 2-Party Payments with applications other than web stores, an Internet connection is still required to interact with the Payment Server.

## integrating 3-party payments

This section describes the information flow and integration model for 3-Party Payments. The 3 parties involved in a 3-Party transaction are the merchant, the payment provider and the cardholder.

3-Party Payments allow the Payment Server to manage the payment pages and collect the cardholder's card details on your behalf. The cardholder's Internet browser is redirected to take the Transaction Request to the Payment Server to process the transaction. After processing the transaction, the cardholder's Internet browser is returned to a web page that you nominate in the transaction together with a Transaction Response. The Transaction Response processing of the receipt information finalises the transaction.

### 3-party payment information flow



The following is the information flow in a 3-Party transaction:

- 1 A cardholder browses your online store, selects a product and enters their shipping details into the merchant's online store at the checkout page.
- 2 The cardholder clicks a pay button and your online store sends the payment request to the Payment Server by redirecting the cardholder's Internet browser to the Payment Server.
- 3 The Payment Server prompts the cardholder for the card details using a series of screens.

The first screen displays the cards supported, for example MasterCard, Visa, and American Express. The cardholder chooses the card type they want to use for the transaction.

The second screen accepts the details for the chosen card such as card number, card expiry, and card security number if required.

- 4 The Payment Server passes the details to the acquirer bank to process the transaction. After processing, the Payment Server displays the result of the transaction with a receipt number if it was successful or an appropriate information message if it was declined. It then advises the cardholder to wait while they are redirected back to the merchant's site.
- 5 The Payment Server then redirects the cardholder back to merchant's site with the transaction response. The response contains the result of the transaction.
- 6 The online store interprets the response and displays the receipt and confirms the order to the cardholder for their records.

## what the cardholder sees

In 3-Party Payments, the cardholder is presented with the following pages:

- 1 The merchant's application checkout page.

QTY	Code	Brand	Description	Size	Unit Price	Total Price
1	Y224501	SPLIT	Jerome	Small	\$69.95	\$69.95
1	FREIGHT		Express Air Freight		\$20.00	\$20.00
<b>Total:</b>						<b>\$89.95</b>

**ENTER YOUR DELIVERY ADDRESS**

Name:

Delivery Address:

Delivery City:

Delivery State:

Zipcode:

Country:

The application checkout page displays the line items that the cardholder wants to purchase and the total amount to pay, including any delivery charges and taxes. The cardholder accepts the amount and proceeds to the payment server payment pages to enter their card details. The application checkout page is created by the merchant and displayed on their website.

## 2 The Payment Server's payment options page.

The Payment Server creates this and the following pages.

The screenshot shows a web page titled "Select your preferred payment method" under the "TEST MODE" header. The merchant name is "adam". Below the title, it says "Pay securely using SSL+ by clicking on the card logo below:". There are six logos displayed: DISCOVER, DELTA, VISA, MasterCard, Diners Club International, and AMERICAN EXPRESS. A "Cancel" link is at the bottom. The footer includes "Copyright ©2007 Dialect Payments Pty Ltd. All Rights Reserved." and "SECURE PAYMENTS POWERED BY DIALECT".

The payment options page presents the cardholder with the card types the merchant accepts. The cardholder clicks a card type and proceeds to the Payment Details web page.

## 3 The Payment Server's payment details page.



The screenshot shows a web page titled "Enter your card details" under the "TEST MODE" header. The merchant name is "adam". Below the title, it says "MasterCard: You have chosen MasterCard as your method of payment. Please enter your card details into the form below and click 'pay' to complete your purchase." The form includes fields for Card Number (with a red box highlighting the last four digits), Expiry Date (month/year), and Security Code (with a red box highlighting the last three digits). Below the Security Code field, there is an image of a MasterCard card with a red box highlighting the last four digits of the card number. The Cardholder Name field is labeled "Please enter your name as it appears on the card." The Purchase Amount is "AUD \$10.00". There is a "Cancel" button and a "pay" button. At the bottom, there is a line for the cardholder to sign: "I hereby authorise the debit to my MasterCard Account in favour of adam". The footer includes "Copyright ©2007 Dialect Payments Pty Ltd. All Rights Reserved." and "SECURE PAYMENTS POWERED BY DIALECT".

On the Payment Details page, the cardholder enters their card details including the card number, expiry date, card security code (if applicable), and the cardholder name. Then the Payment Server processes the payment.

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



#### 4 The Payment Server's payment pending page.

As the bank is processing the payment, a payment pending page can be displayed to the cardholder.



TEST MODE	
Merchant name: adam	
	<b>Please wait while your payment is processed</b>
<b>Please wait...</b>	
The server is processing your payment using MasterCard for the value of AUD \$10.00.	
Copyright ©2007 Dialect Payments Pty Ltd. All Rights Reserved.	
SECURE PAYMENTS  POWERED BY DIALECT	

#### 5 The Payment Server's redirection page.

The redirection page is displayed in the cardholder's browser and the Transaction Response is passed to your application. A successful transaction would appear as follows:

TEST MODE	
Merchant name: adam	
	<b>Transaction Results</b>
<b>Result of your transaction:</b>	
Your payment has been <b>approved</b> .	
Your receipt number is: <b>080903000018</b>	
Please wait while you are redirected back to the merchant..	
Copyright ©2007 Dialect Payments Pty Ltd. All Rights Reserved.	
SECURE PAYMENTS  POWERED BY DIALECT	

Or, if declined, the following page would appear:

TEST MODE	
Merchant name: adam	
	<b>Transaction Results</b>
<b>Result of your transaction:</b>	
Your payment was NOT successful. The transaction was declined by the acquirer.	
Please wait while you are redirected back to the merchant..	
Copyright ©2007 Dialect Payments Pty Ltd. All Rights Reserved.	
SECURE PAYMENTS  POWERED BY DIALECT	

6 The merchant's application receipt page.

**UNIVERSAL**  
ONLINE STORE

1 START ORDER 2 SELECT DELIVERY METHOD 3 ENTER DELIVERY ADDRESS 4 PAY 5 FINISH ORDER

**UNIVERSAL STORE RECEIPT**  
Print a copy of this receipt for your future reference. [Print Receipt](#)

**APPROVED** **Your purchase payment was approved!**  
We now need the delivery address for your purchase. Simply complete the form below and press 'continue' to finalize your purchase.

Name: Invoice Number: 2760  
Delivery Address: Date of purchase: 16/06/01  
City:  
State: CALIFORNIA  
Postcode: 1234  
Country: United States

**YOUR PURCHASES**

QTY	Code	Brand	Description	Size	Unit Price	Total Price
1	Y234501	SPLIT	Jerome	Small	\$69.95	\$69.95
1	FREIGHT		Express Air Freight			\$20.00
<b>Total:</b>					<b>\$89.95</b>	

[Continue](#)

The application receives the Transaction Response and displays a receipt page. The application receipt page is created by you and displayed on your website.



## integrating 3-party payments with virtual payment client

To process a payment your application needs to be integrated with the Virtual Payment Client in order to:

- Create the MD5 signature
- Send it with the Transaction Request
- Check the MD5 signature in the Transaction Response is valid for the received data

To do this you need to do the following:

### handle a transaction request

- 1 Add any variables required by the application to re-create the session before the Transaction Request has been processed. These should be included in the MD5 signature.
- 2 Collect the minimum required information for a Transaction Request. This includes your MerchTxnRef, Merchant ID, an Order Information field, the Transaction Amount, the Locale and the Return URL to which the Payment Server needs to redirect the cardholder.

You may require additional information fields when using optional features.

- 3 Formulate a Transaction Request using the fields outlined above.
- 4 Redirect the cardholder's Internet browser using the Transaction Request you just created. At this point the cardholder session with your application is interrupted while the cardholder is redirected to the Payment Server.

An example of the start of a Transaction Request is:

```
https://Virtual_Payment_Client_URL/vpcpay?vpc%5FVersion=1&vpc%5FLocale=en&vpc%5FCommand=pay&vpc%5FAccessCode=A8698556&vpc%5FMerchTxnRef=123&vpc%5FMerchant=TESTMERCHANT&vpc%5FOrderInfo=Example&vpc%5FAmount=100&vpc%5FReturnURL=http%3A%2F%2FMerchant_Web_URL%
```

## what the payment server does

When a Transaction Request arrives at the Payment Server, it:

- Checks the digital signature on the Digital Receipt, and if correct it decrypts the encrypted Digital Receipt data and the Payment Server:
  - Displays the card selection page for the cardholder to choose their card type.
  - Displays the card details page so that the cardholder can provide the card details for the selected card type.
  - Processes the data and sends the Transaction Response to the acquiring bank so that the funds can be settled into the merchant's account.
  - Sends back a Transaction Response to the website page (as nominated by the ReturnURL in the Transaction Request) indicating whether the transaction was successful or declined. The Payment Server generates a signature hash that is sent with the data.
  - The Payment Server can also send back error messages, if for example there is a communication error in the banking network and the transaction cannot proceed.
- If the MD5 Signature is incorrect, the Payment Server:
  - Returns the cardholder back to the merchant with a Transaction Response with an error message indicating the MD5 signature was invalid in the Transaction Request. No payment takes place.

## handle a transaction response

The Transaction Response is returned to your website using an Internet browser redirect as specified in the `vpc_ReturnURL` field. The DR will always have a secure hash for the online store to check data integrity. An example of the start of a transaction response is:

```
http://Merchant_Site_URL/Receipt.asp?vpc_AVSErrorCode=Unsupported&vpc_AcqAVSRespCode=
Unsupported&vpc_AcqCSCRespCode=Unsupported&vpc_AcqResponseCode=00&vpc_Amount=100&vpc_A
uthorizeId=020072&vpc_BatchNo=20051209&vpc_CSCResultCode=Unsupported&vpc_Card=MC&vp
```

The merchant application receipting function needs to be able to calculate the MD5 signature in the Transaction Response to determine if the signature received is valid for the receipt data. It has to handle:

- Incorrect MD5 Signatures.
- Successful transactions.
  - If **vpc\_TxnResponseCode** code is equal to '0' then the transaction was completed successfully and you can display a receipt to the cardholder.
- Declined transactions.
  - If **vpc\_TxnResponseCode** is equal to '1', '2', '3', '4', or '5' the transaction has been declined and this needs to be conveyed back to the cardholder.

- Error Conditions.
  - If **vpc\_TxnResponseCode** equals '7' or '8' an error has occurred.
  - Other values may indicate an error has occurred.
  - Further details for error conditions can be gathered by examining the **vpc\_Message** field so a decision can be made as to the next step.

All four of these conditions are responses that can occur back from the Virtual Payment Client.

## handle session variables

A session begins when a cardholder enters your website and ends when they leave your website.

Some merchant applications use session variables to keep track of where the merchant application is up to and to prevent unauthorised entry without the cardholder signing in. This stops hackers from spoofing transactions.

Other applications create session variables in other ways. Some applications do not create session variables at all. If there are no session variable(s) in your application then the next section will not apply.

## sending session variables to the payment server

When using 3-Party Payments, the Virtual Payment Client requires the cardholder's browser to support cookies. In 3-Party Payments, the cardholder browser's connection is completely severed from the merchant's application.

Session variables that are required to identify the users must be collected and sent to the Payment Server. The session variables are not used by the Payment Server, but are returned appended to the Transaction Response.

You can store up to 5 session variables using any name that your application needs, providing:

- They conform to HTTP/HTTPS protocols. To make them conform to the standard URL, you need to URL encode all session variables before sending them.
- A URL can only be a maximum total of 2047 characters long; otherwise the browser will not perform a redirect function.
- Their names must not start with **vpc\_**. These variables must be present in the Transaction Request before the MD5 signature is calculated and appended to it.

The session variables are not stored in the Payment Server database. The Virtual Payment Client will send these session fields back to the merchant application in the Transaction Response. This allows the merchant application to recover the session variables from the Transaction Response, and use them to restore the session. The session then continues as though it had never been broken.

## additional 3 party functionality

The Payment Server provides you with additional ways to process payments, for example:

- 3-Party, where the merchant collects the cardholder's card type
- 3-Party, where the merchant collects all the cardholder's card details
- 3-Party, where the merchant is enabled for Verified by Visa™ and MasterCard SecureCode™. For more information, see **Using 3-D Secure Authentication** (see "Using 3-D Secure Payment Authentications" on page 23).

### 3-party payments where the merchant collects the cardholder's card type

In 3-Party Payments you can choose to bypass the Payment Server payments page that displays the logos of all the cards the Payment Provider will accept. This can be helpful if your 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 your application and once on the Payment Server.

You can achieve this by providing the following extra fields in the Transaction Request – Gateway and Card Type. This type of 3-Party transaction is called External Payment Selection (EPS). You must have the EPS privilege enabled in your merchant profile if you wish to use this functionality. Contact your Payment Provider to have this privilege enabled.

### 3-party payments where the merchant collects all the cardholder's card details

In 3-Party Payments you can choose to bypass all the Payment Server payment pages displayed. This can be helpful if you want to keep merchant branding consistent throughout a 3-Party transaction. The same security measure must be observed, such as installing an SSL certificate to protect the card details being sent from the cardholder's browser to the merchant's site as in a 2-Party transaction.

You can achieve this by providing the following card details in extra fields in the Transaction Request. These fields are: Card Number, Card Expiry, Gateway and Card Type. You can also submit Card Security Code, if available and any other optional data at this point.

You must have the EPS, Card Details and 3-Party privileges enabled in your merchant profile if you wish to use this functionality. Contact your Payment Provider to have these privileges enabled.

## 3-party payments using Verified by Visa™ and MasterCard SecureCode™

In 3-Party Payments you can have your Payment Provider enable you to use Verified by Visa™ and MasterCard SecureCode™, which provides additional security to all your payments. This type of transaction requires the 3-Party model, and works by redirecting the cardholder to their card issuer to enter a password. For more information, please see ***Using 3-D Secure Authentications to secure your payments*** (see "Using 3-D Secure Payment Authentications" on page 23).

Verified by Visa™ and MasterCard SecureCode™ can be implemented using a standard 3-Party transaction; a 3-Party transaction where you bypass the card selection page (EPS); or a 3-Party transaction where the merchant supplies full card details.

You can choose with this last transaction type to bypass all the Payment Server payment pages displayed. This can be helpful if you want to keep merchant branding consistent throughout a 3-Party Verified by Visa™ and MasterCard SecureCode™ transaction, except for the one page where the cardholder types in their password. The same security measure must be observed, such as installing an SSL certificate to protect the card details being sent from the cardholder's browser to the merchant's site as in a 2-Party transaction.

You can achieve this by providing the following card details in extra fields in the Transaction Request. These fields are: Card Number, Card Expiry, Gateway and Card Type. You can also submit Card Security Code and any other optional data at this point.

You must have the EPS, VbV and 3-Party privileges enabled in your merchant profile if you wish to use this functionality. Contact your Payment Provider to have these, and any other 3-party privileges enabled.

## supplementary transactions

To implement the supplementary features available on the Payment Server, you need to add additional data to the Transaction Request.

Multiple supplementary features may be combined in the one Transaction Request, but you need to ensure that the functionality being implemented can be combined. See **Advanced Function Compatibility** (see "Advanced Function Compatibility" on page 71).

Some additional data returned in the Transaction Response can be accessed using the appropriate key value.

### address verification service (AVS)

Address Verification Service (AVS) is a security feature used for card not present transactions that compares the billing address entered by the cardholder with the records held in the card issuer's database. An AVS result code is returned in the transaction response message indicating the extent to which the addresses match (or fail to match). The merchant application is responsible for deciding how to handle the payment transaction on the basis of the AVS result code.

**Note:** AVS is only supported for the S2I Acquirer. Furthermore, not all banks support AVS, so even though AVS data is passed in the transaction data, if the issuing bank does not support AVS, it will be ignored.

### card holder name transactions

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

**Note:** Applies only to 3-party transactions.

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

## card security code (CSC/CCV2)

The Card Security Code (CSC) is a security feature used for card not present transactions that 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.



For American Express, the number is the 4-digit value printed on the front above the credit card account number.



In a 2-Party transaction and a 3-Party with card details the card security code is sent, but in a standard 3-Party transaction the Payment Server requests this information from the cardholder. Depending on the level of the match between what the cardholder issuing institution has on file and what the cardholder has provided in the transaction, determines if the transaction will complete successfully or fail.

In most cases if the transaction fails due to the CSC not being accepted, it results in a declined transaction with **vpc\_TxnResponseCode** = "2" – 'Bank Declined Transaction'

For some Payment Providers, the CSC result code (*CSCResultCode*) is returned, which indicates the level that the CSC code matches the data held by the cardholder issuing institution. However this is not always provided and may show up as 'Unsupported'.

**Note:** If CSC is collected by the merchant, this data is never to be stored or retained. This includes storing it in a logfile.

CSC is mandatory in some countries and regions. Please check with your Payment Provider to determine the legal requirements of your region.

However, not all banks support CSC so even though the data is passed in the transaction data, if the issuing bank does not support CSC, it will be ignored.

## card present transactions

This feature allows merchants to add Card Present information and track data to a transaction. This feature applies where the merchant integration collects card track data from POS terminals. Card present functionality can only be performed as a 2-Party Authorisation/Purchase transaction.

The card track data needs to contain the correct start and end sentinel characters and trailing longitudinal redundancy check (LRC) characters.

For all card present transactions, the Merchant Transaction Source must be set to the value **'CARDPRESENT'**.

Regarding card track data:

- If both are available, both *vpc\_CardTrack1* and *vpc\_CardTrack2* must be added to the Transaction Request.
- If only one is available, either *vpc\_CardTrack1* or *vpc\_CardTrack2* must be added to the Transaction Request.

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 Capability*' values in *vpc\_POSEntryMode* field to indicate that the card was sighted, but manually entered.

**Note:** Card Track 3 data is not supported.

For EMV transactions, **'CARDPRESENT'** is used. The other mandatory fields are: *EMVICCData*, *vpc\_CardSeqNum*, *vpc\_POSEntryMode*, and *vpc\_CardTrack2*. Card types must be MasterCard or Visa.

## external payment selection (EPS)

EPS is only used in a 3-Party transaction for bypassing the Payment Server page that displays the logos of all the cards the payment processor will accept. 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.



## merchant transaction source

Merchant transaction source functionality allows a merchant to indicate the source of a 2-Party transaction. These can be:

- 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.

This can only be used if the merchant has their privilege set to use this functionality; otherwise the transaction will be set to the merchant's default transaction source as defined by the Payment Provider.

- VOICERESPONSE – indicates that the merchant has captured the transaction from an IVR system.

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 frequency

Merchant transaction frequency functionality allows a merchant to set the frequency of the transactions for the cardholder's order.

This can only be used if the merchant has their privilege set to use this functionality; otherwise the transaction will be set to the merchant's default transaction source as defined by the Payment Provider.

The frequencies are:

- 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:** The Payment Server does not contain a recurring payment facility to automatically trigger a recurring payment. This is up to the merchant to implement.

## referral message

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** (see "Referral Transaction" on page 44).

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

## referral transaction

Referral transactions allow a merchant to resubmit a referred initial transaction with an authorisation code obtained from the Issuer. However, the amount cannot be altered in this transaction.

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. This transaction must always follow the referred transaction. See **Referral Message** (see "Referral Message" on page 44).

**Note:** Applies only to 2-party transactions.

## airline ticket number

Ticket Number functionality allows the merchant to enter additional information in the Transaction Request about the transaction that will be stored in the Payment Server database. Although the ticket number was originally designed for the travel industry, it can be any alphanumeric data about the transaction up to 15 characters.

It is available for both 2-Party and 3-Party transactions. The ticket number is returned in the Transaction Response and is passed to the financial institution as part of certain transactions.

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

## risk management

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.

The solution introduces various rules for risk mitigation — IP Country, Card BIN (Bank Identification Number), Trusted Cards, Suspect Cards, 3-D Secure, IP Address Range — each rule contributes differently to the risk profile. IP Address Range and Card BIN rules enable blocking transactions from high-risk IP address ranges and high-risk BIN ranges respectively. Trusted Cards and Suspect Cards allow you to create lists of trustworthy card numbers and suspected card numbers respectively. 3DS rules enable you to block transactions based on authentication states and IP Country rules enable you to block countries with high-risk IP addresses.

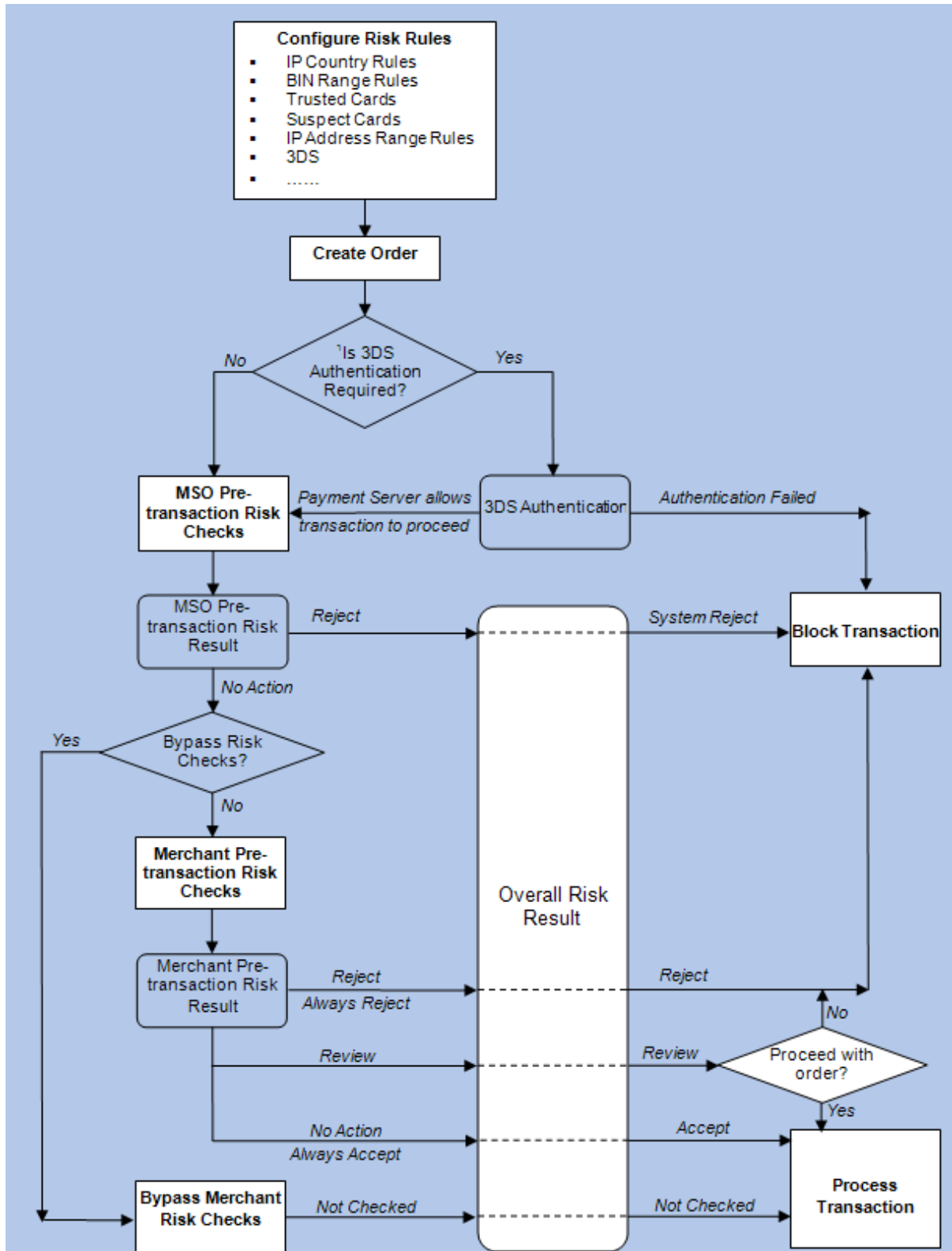
Rules can be configured at both the merchant level and MSO level; however, Suspect Cards and Trusted Cards can be configured at the merchant level only. MSOs have the added advantage of defining rules that merchants cannot bypass — MSO rules always override merchant rules. Also, an MSO rule configured to reject a transaction has the ability to not only block the transaction but also block merchant configured rules from being processed. MSOs, however, cannot configure rules for review unlike merchants who can configure rules for reject, review, or normal processing of a transaction.

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

**Note:** Risk Management is applicable only to:

- Merchants who have *May Use Risk Management* privilege enabled.
- Transaction modes, *Auth Then Capture* and *Purchase*. Standalone Captures, Standalone Refunds, etc., will not be assessed for risk.

The diagram below illustrates the process flow of an order when risk management is enabled for an MSO/merchant.



## bank account type

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.

This identifier is mandatory if the card type is Maestro, and will be displayed in the Order Search results in the Merchant Administration portal on the Payment Server.

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

## payment authentication

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

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

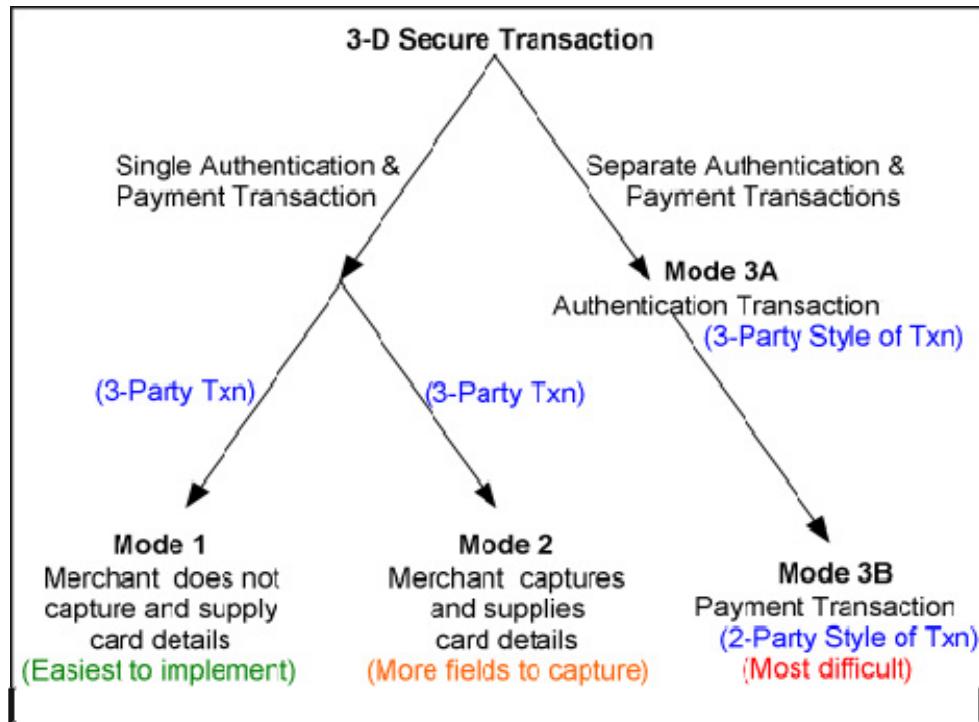
During a transaction, 3DS authentication allows the merchant to authenticate the cardholder by redirecting them to their card issuer where they enter a previously registered password.

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

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

## payment authentication 3-D Secure transaction modes

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



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

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

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

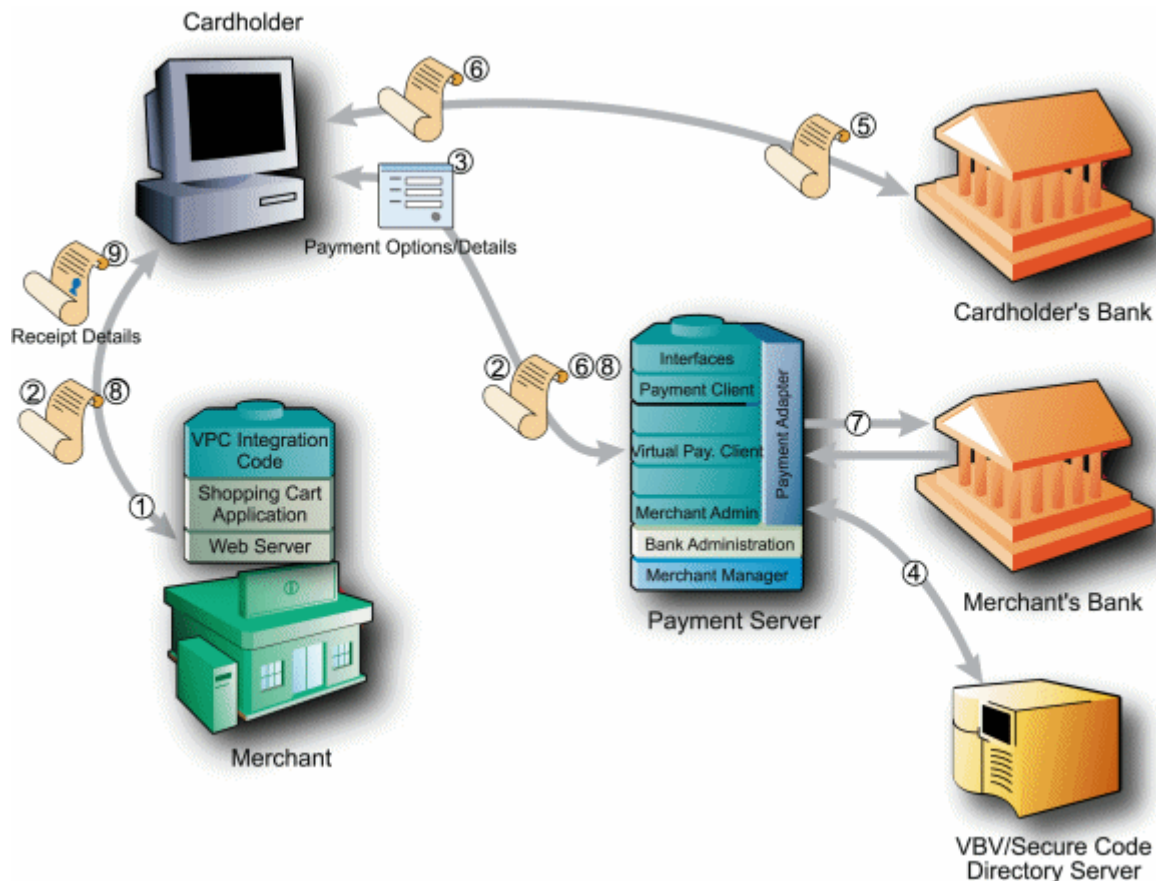
The *merchant's application* collects the cardholder's card details and sends them to the Payment Server, which redirects the cardholder to the card-issuing bank to enter their 3-D Secure password. If the Authentication is performed correctly the Payment Server uses the Authentication information to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

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

The Authentication operation outputs become the inputs for a 3-Party with card details transaction.

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

## information flow of a 3D-Secure authentication/payment transaction



If you have been enabled to use Verified by Visa™ and MasterCard SecureCode™, the information flow for Verified by Visa™ and MasterCard SecureCode™ where the Payment Server collects the card details (Mode1) is as follows:

- 1 A cardholder browses the application, selects a product and enters their shipping details into the merchant's application at the checkout page.
- 2 The cardholder clicks a pay button and your application sends the payment Transaction Request to the Payment Server by redirecting the cardholder's Internet browser to the Payment Server.
- 3 The Payment Server prompts the cardholder for the card details.
- 4 If the card is a Visa or MasterCard, the Payment Server then checks with the VBV or SecureCode Directory Server to determine if the card is enrolled in either the Verified by Visa™ (Visa 3-Domain Secure) or MasterCard SecureCode™ (MasterCard 3-Domain Secure) scheme.

If the card is not enrolled in payment authentication scheme then go to Step 7.

If the cardholder's card is registered in the payment authentication scheme, the Payment Server redirects the cardholder's browser to the card issuing site for authentication.



- 5 If the cardholder's card is registered in the payment authentication scheme, the Payment Server redirects the cardholder's browser to the card issuer's site for authentication. The card issuer's server displays the cardholder's secret message and the cardholder enters their secret password, which is checked against the Issuing bank's database.
- 6 At the completion of the authentication stage, the cardholder is redirected back to the Payment Server indicating whether or not the cardholder's password matched the password in the database.  
  
If the cardholder was not authenticated correctly, then the payment does not take place and the cardholder is redirected back to the merchant's site with a Transaction Response containing details to indicate the authentication failed – see step 8.
- 7 If the cardholder was authenticated correctly, or Payment Authentication did not occur the Payment Server continues with processing the transaction with the results of the authentication attempt.
- 8 The Payment Server then redirects the cardholder back to merchant's site with the Transaction Response. The Transaction Response contains the result of the transaction.
- 9 The application processes the Transaction Response and displays the receipt.

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

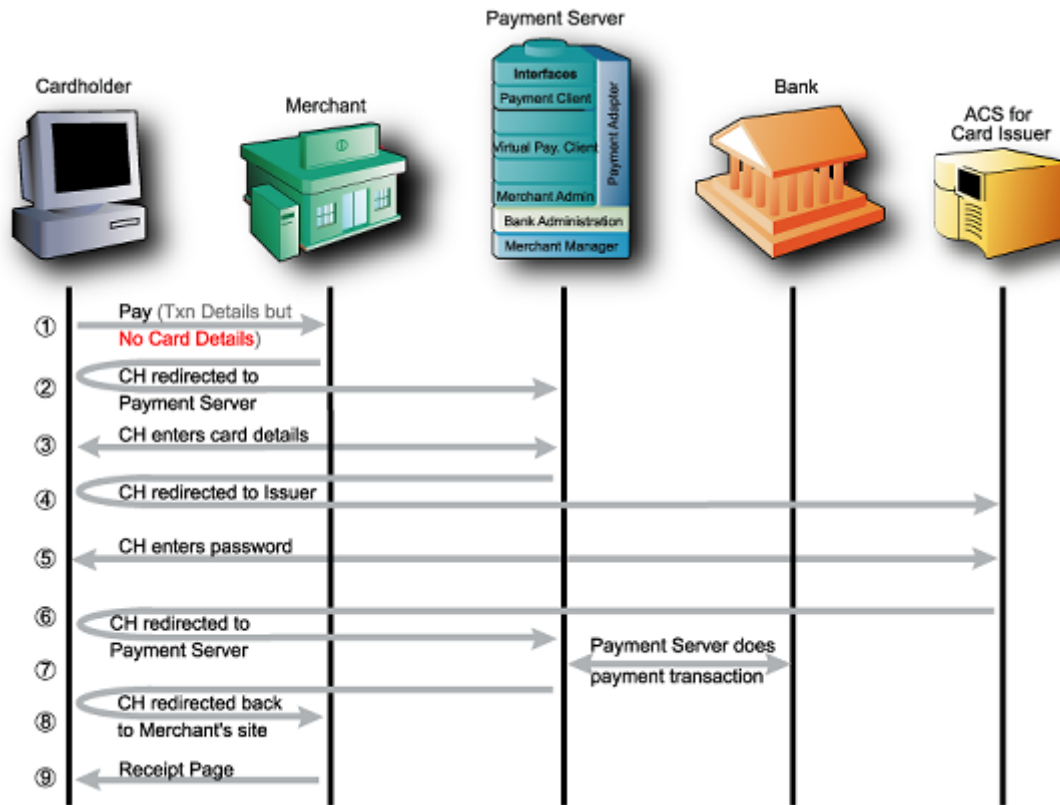
Mode 2 and Mode 3a are slight variations on the above information flow. In mode 2 and mode 3a the merchant collects the card details and passes them through, which means step 3 is eliminated.

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

## advantages and disadvantages of the 3-D Secure modes of transaction

Mode	Advantages	Disadvantages
<b>Mode 1</b> 3 Party Authentication and Payment transaction mode	<ul style="list-style-type: none"> <li>Simple to implement.</li> <li>The Payment Provider collects the cardholder's card details and not the merchant, which provides highest level of security for the cardholder's card details.</li> </ul>	<ul style="list-style-type: none"> <li>The merchant is not able to use their own branding throughout the whole transaction, as the Payment Provider displays their own branding while the card details are being captured.</li> <li>If the cardholder is not enrolled in 3-D Secure, or the authentication could not be performed, the authentication will not take place and the transaction will automatically move into the payment stage.</li> </ul>
<b>Mode 2</b> 3 Party Authentication and Payment transaction (Merchant collects card details)	<ul style="list-style-type: none"> <li>Suits a merchant that normally collects all the card details.</li> <li>Branding of the payment pages on the website remains consistent throughout the whole transaction, except the screen where the cardholder enters their password for 3-D secure.</li> </ul>	<ul style="list-style-type: none"> <li>If the cardholder is not enrolled in 3-D Secure the authentication will not take place and the transaction will automatically move into the payment stage.</li> </ul>
<b>Mode 3a</b> 3 Party Authentication Only transaction mode	<ul style="list-style-type: none"> <li>Suits a merchant that normally collects all the card details.</li> <li>Branding of the payment pages on the website remains consistent throughout the whole transaction, except the screen where the cardholder enters their password for 3-D secure.</li> </ul>	<ul style="list-style-type: none"> <li>It consists of two separate transactions, the Authentication and the Payment, which can be more difficult for a merchant to integrate.</li> </ul>
<b>Mode 3b</b> 2 Party Pre-Authenticated transaction mode	<ul style="list-style-type: none"> <li>Gives the merchant maximum control of the transaction. If the cardholder is not enrolled in 3-D Secure, then the merchant's application can stop the transaction from progressing to the Payment stage providing full control over the transaction risk.</li> <li>Branding remains consistent throughout the whole transaction, except for the one screen where the cardholder enters their 3-D Secure password.</li> </ul>	<ul style="list-style-type: none"> <li>Can only be performed if the merchant collects all the card details.</li> <li>It consists of two separate transactions, the Authentication and the Payment, which can be more difficult for a merchant to integrate.</li> </ul>

## mode 1 – implementing a 3 party authentication and payment transaction (payment server collects card details)



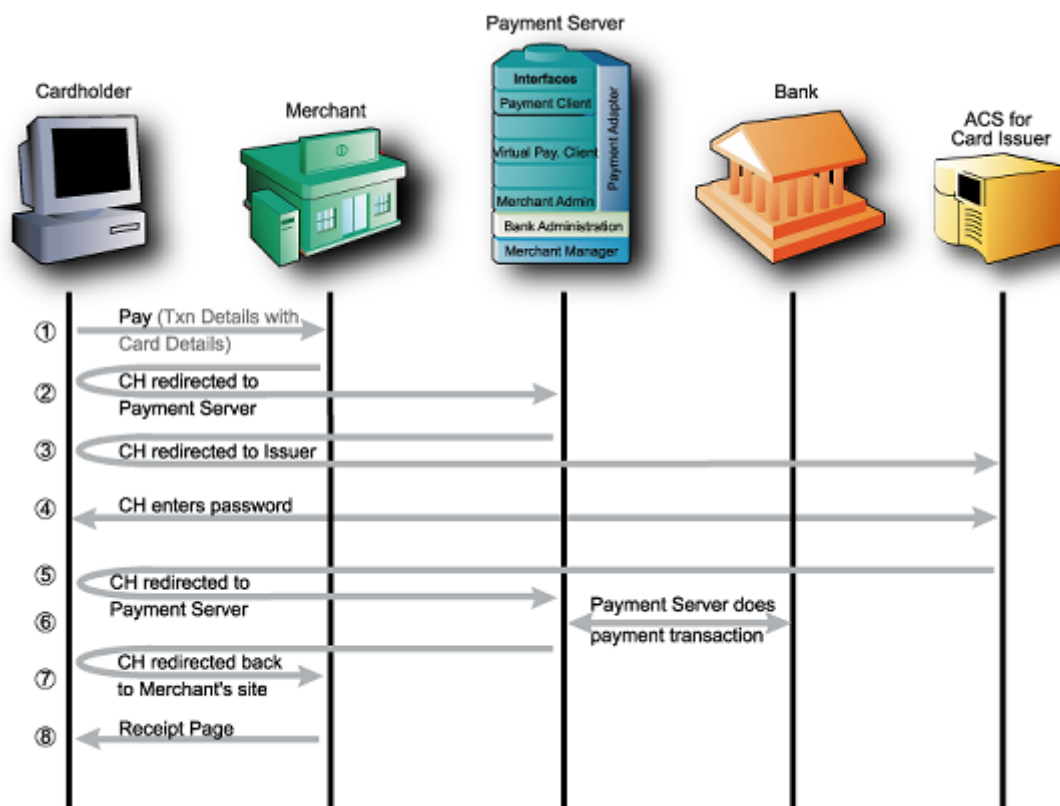
The information flow for a successful 3 Party Authentication and Payment transaction mode is very similar to a standard 3-Party transaction.

- 1 The cardholder submits the order.
- 2 The browser is redirected to the Payment Server.
- 3 The Payment Server collects the cardholder's card details and determines if the cardholder is enrolled in 3-D Secure. If the card is not enrolled in 3-D Secure then steps 4, 5 and 6 are skipped.
- 4 The Payment Server redirects the cardholder's browser to the Access Control Server (ACS) site for authentication.
- 5 The ACS displays the cardholder's secret message and the cardholder enters their password, which is checked with the Card Issuer system.
- 6 The cardholder is redirected back to the Payment Server and the card issuer returns an authentication message showing whether or not the cardholder's password matched the password in the card issuer system. If the Authentication failed, step 7 is bypassed and the cardholder is redirected back to the merchant (see step 8) with a **vpc\_TxnResponseCode** of 'F'. No payment takes place in this scenario.

- 7 If the cardholder is authenticated correctly, the Payment Server continues with processing the payment part of the transaction. The cardholder is redirected back to the merchant, where the receipt is passed back to the merchant.
- 8 The receipt displayed to the cardholder.

## mode 2 – implementing a 3 party authentication and payment transaction (merchant collects card details)

The information flow for Mode 2 transaction where the merchant collect the card details uses the basic 3-Party style of transaction with some additional input fields. For more information, please refer to Basic 3 Party Transaction.



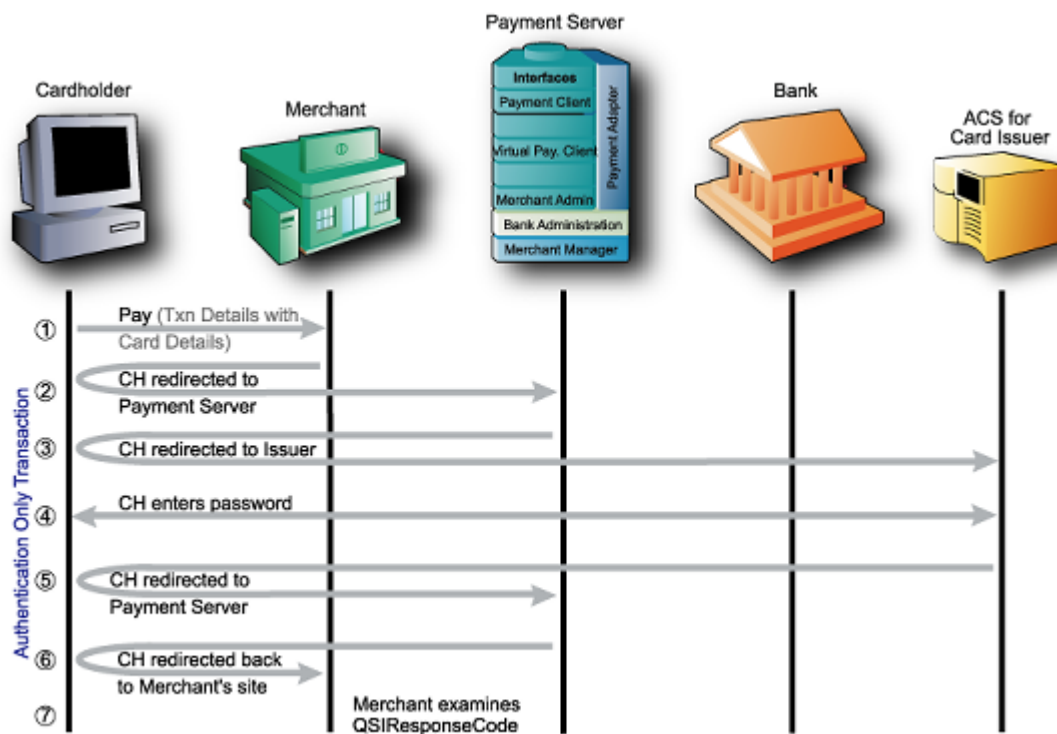
The information flow for a 3 Party Style Authentication & Payment mode 2 transaction is:

- 1 The cardholder enters their card details into the merchant's application, clicks the merchant's Pay button.
- 2 Their browser is redirected to the Payment Server. The Payment Server determines if the card is enrolled in the 3-D Secure scheme by checking the Verified by Visa™ and MasterCard SecureCode™ system. If the card is not enrolled in 3-D Secure then steps 3, 4 and 5 are skipped.
- 3 If the cardholder's card is registered, the Payment Server redirects the cardholder's browser to the Access Control Server (ACS) site for authentication.

- 4 The ACS displays the cardholder's secret message; the cardholder enters their secret password, which is checked with the Card Issuer system.
- 5 The cardholder is redirected back to the Payment Server and the card issuer returns an authentication message showing whether or not the cardholder's password matched the password in the card issuer system. If the Authentication failed, step 6 is bypassed and the cardholder is redirected back to the merchant (see step 7) with a **vpc\_TxnResponseCode** of 'F'. No payment takes place in this scenario.
- 6 If the cardholder is authenticated correctly, the Payment Server continues with processing the payment part of the transaction.
- 7 The cardholder is redirected back to the merchant, where the receipt is passed back to the merchant.
- 8 The receipt displayed to the cardholder.

## mode 3a – implementing a 3 party style authentication only transaction

In a 3 Party Style Authentication Only transaction mode, the merchant's application can stop the transaction from progressing to the Payment stage and return an error message back to the cardholder if the cardholder is not enrolled in 3-D Secure. The other scenario is a merchant may want to perform an immediate Authentication operation, but delay the payment transaction. The information flow for a 3 Party Style Authentication Only transaction mode uses the 3-Party style of transaction with additional card details. For more information, please refer to Basic 3 Party Transaction.



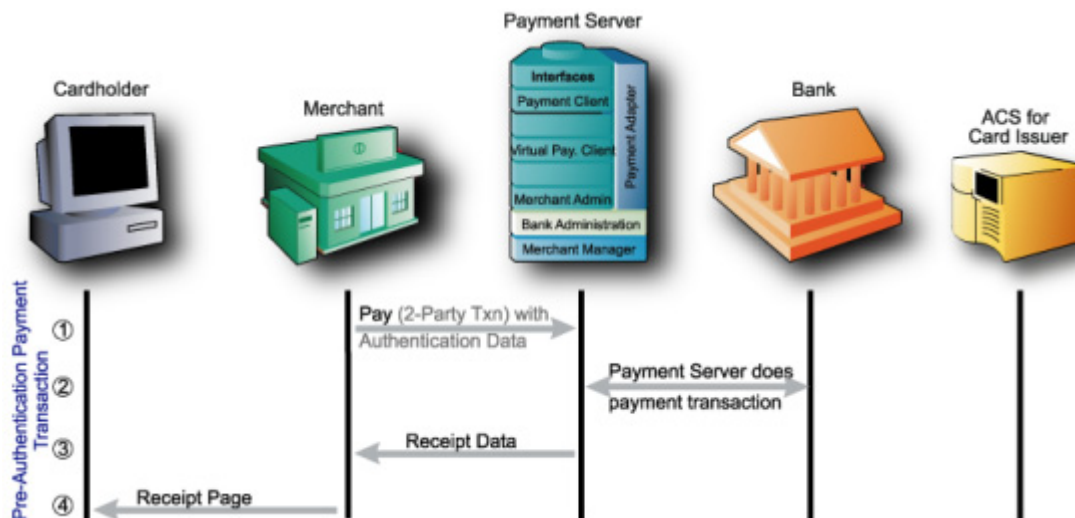
The information flow for a 3 Party Style Authentication Only mode transaction is:

- 1** The cardholder enters their card details into the merchant's application.
- 2** Their browser is redirected to the Payment Server and the Payment Server determines if the card is enrolled in the 3-D Secure scheme by checking the Verified by Visa™ and MasterCard SecureCode™ system.
- 3** If the cardholder's card is registered, the Payment Server redirects the cardholder's browser to the Access Control Server (ACS) site for authentication.  
If the card is not enrolled in 3-D Secure then steps 3, 4 and 5 are skipped.
- 4** The ACS displays the cardholder's secret message; the cardholder enters their secret password, which is checked with the Card Issuer system.
- 5** The cardholder is redirected back to the Payment Server and the card issuer sends an authentication message showing whether or not the cardholder's password matched the password in the Card Issuer system.
- 6** The cardholder is redirected back to the merchant's site.
- 7** The merchant examines the **vpc\_TxnResponseCode** to determine if the cardholder was enrolled, and if they were successfully authenticated.

The merchant can now determine whether or not to proceed with the Payment. The Authentication outputs from this transaction would then be used for the Pre-Authenticated transaction along with the card details.

## mode 3b – implementing a 2-party style pre-authenticated payment transaction

The information flow for a 2-Party Style Pre-Authenticated Payment transaction uses the 2 Party style of transaction. For more information on 2 Party transactions, see **2-Party** (see "Integrating 2-Party Payments" on page 28).



The information flow for a 2-Party Style Pre-Authenticated Payment transaction is:

- 1 The authentication outputs from the Authentication Only transaction or external MPI provider are used in the Standard 2-Party transaction with additional input fields.
- 2 The Payment Server then performs the payment part of the Pre-Authenticated Payment transaction.
- 3 The Result of the payment is sent back to the merchant.
- 4 The merchant displays a receipt page to the cardholder, which indicates whether the transaction was successful or not.

## advanced merchant administration (AMA)

There are a number of additional transactional options that you can implement, depending on your implementation of Virtual Payment Client. All of these transactions operate using the 2-Party model.

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

### capture

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

A merchant that operates using Authorisation/Capture mode performs two transactions to transfer the funds into their account.

- 1 The first transaction (Authorisation) reserves the funds on the cardholder's credit card.
- 2 The second transaction (Capture) transfers the funds from the cardholder's account to the merchant's account.

Capture allows a merchant to complete a transaction performed using the Authorisation/Capture Payment Model. The capture transaction initiates the transfer of funds from the cardholder's account to the merchant's account.

**Note:** In Purchase mode, the authorisation and capture operations are completed at the same time in the one purchase transaction, so you do not need to perform a separate capture is not needed, that is this capture command is not necessary if the merchant is operating in Purchase mode.

There are two ways you can capture the funds from an authorisation transaction:

- 1 Manually using Merchant Administration. This is the simplest method if you do not have many transactions. For more information, refer to your *Merchant Administration User Guide*.
- 2 Using the Capture command using the Virtual Payment Client to directly perform the capture transaction from your application.

Payment Providers allow merchants to perform as many capture transactions on the original Authorisation transaction as required, but the total amount captured cannot be more than the amount specified in the original Authorisation transaction, unless the excessive capture privilege is enabled.



## standalone capture

A Standalone Capture allows you to capture funds for an order that was authorised either manually, or in an external system. When performing a Standalone Capture, the externally produced Authorisation ID must be included in the request.

## refund

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.

Refunds can only be performed for a previously completed a purchase or capture transaction for the particular order. The merchant can run any number of refund transactions on the original transaction, but cannot refund more than has been obtained via a purchase or capture transaction.

There are two ways to refund the funds:

- 1 Manually using Merchant Administration. This is the simplest method if the merchant does not have many refund transactions. For more information, refer to your *Merchant Administration User Guide*.
- 2 Using the AMA Refund command using the Virtual Payment Client to directly perform refunds from the merchant's application.

## standalone refund

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.

## void capture

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

This command cannot be used if the merchant is operating in Purchase mode.

The merchant can only run one void capture transaction on the original capture transaction, as it completely removes the capture transaction as though it never occurred. A void capture must be run before the batch containing the original capture transaction is processed by the acquiring institution.

There are two ways you can Void Capture the funds:

- 1 Manually using Merchant Administration. This is the simplest method if you do not have many Void Capture transactions. For more information, refer to your *Merchant Administration User Guide*.
- 2 Using the Void Capture command using the Virtual Payment Client to directly perform Void Captures from your application. The merchant must have a user enabled with AMA and Void privileges to use this functionality.

**Note:** Not all financial institutions support void transactions, only those that operate in switch to issuer mode. Please consult with your financial institution if they support voids.

Only the most recent transaction in an order can be voided.

## void refund

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

The merchant can only run one Void Refund transaction on the original refund transaction as it completely removes the refund transaction as though it never occurred. The Void Refund must be run before the acquiring institution processes the batch containing the original refund transaction.

There are two ways you can Void Refund the funds. Your Payment Provider must enable this function on your Merchant Profile for you to use either of these methods:

- 1 Manually using Merchant Administration. This is the simplest method if you do not have many Void Refund transactions. For more information, refer to your *Merchant Administration User Guide*.
- 2 Using the Void Refund command using the Virtual Payment Client to directly perform Void Refund from your application. The merchant must have a user enabled with AMA and Void privileges to use this functionality.

**Note:** Not all financial institutions support void transactions. Please consult with your financial institution if they support voids.

Only the most recent transaction in an order can be voided.

## void purchase

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. This transaction is not possible for Debit and EBT transactions.

The merchant can only run one 'Void Purchase' transaction on the original 'Purchase' transaction as it completely removes the purchase transaction as though it never occurred.

The Admin Void Purchase must be run before the acquiring institution processes the batch containing the original purchase transaction.

There are two ways you can Void Purchase the funds. Your Payment Provider must enable this function on your Merchant Profile for you to use either of these methods:

- 1 Manually using Merchant Administration. This is the simplest method if you don't have many Void Purchase transactions. For more information please refer to your *Merchant Administration User Guide*.
- 2 Using the Void Purchase command using the Virtual Payment Client to directly perform Void Purchase from your application. The merchant must have a user enabled with AMA and Void privileges to use this functionality.

**Note:** Not all banks support void transactions, only those that operate in switch to issuer mode. Consult with your financial institution if they support voids.

Only the most recent transaction in an order can be voided.

## QueryDR

The AMA QueryDR command allows a merchant to search for a duplicate transaction receipt. The search is performed on the key – *vpc\_MerchTxnRef*, so the *vpc\_MerchTxnRef* field must be a unique value.

If you want to use QueryDR to return duplicate 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.

If a transaction receipt is found, the results will contain the same fields as the original receipt plus the 2 flags described below.

QueryDR always returns these 2 flags:

- **vpc\_DRExists:** If no transactions are found that match the *vpc\_MerchTxnRef* number, this value will be set to 'N' for No. If any transactions are found that match the *vpc\_MerchTxnRef* number, this value will be set to 'Y' for Yes.
- **vpc\_FoundMultipleDRs:** This is used to determine if there are multiple results. If the value is "N", then only one *vpc\_MerchTxnRef* matches the search criteria. If the value is "Y", then there are multiple *vpc\_MerchTxnRef* matching the search criteria, but it will return the most recent transaction. If the query result returned is not the correct one, the merchant must manually search through Merchant Administration on the Payment Server.

**Note:** QueryDR does not return receipt data for 3-D Secure (Verified by Visa™ and MasterCard SecureCode™) Authentication Only transactions.

## excessive captures

The Excessive Captures feature enables the merchant to perform captures for amounts greater than the authorized amount. This feature is available for orders performed in MA or through the 2-party gateway.

**Note:** The excess permitted is specified as a percentage of the original authorized amount, and is determined at the level of the MSO. This means that the permitted excess percentage is the same for all merchants associated with a given MSO.

## AMA shopping transaction history

A shopping transaction history returns all of the OrderID reference transactions (shopping transactions) between a start date and an end date where the cardholder entered their card details for a 2-Party or 3-Party transaction. In Purchase mode it would be the Purchase transaction; or the Auth transaction in Auth/Capture mode. It also contains summary information for the entire shopping experience, including captures and refunds.

The Shopping Transaction History command allows a merchant to view the details of shopping transactions for the MerchantId in the Payment Server over a specified period. The period is specified by the use of a Start Date/Time value and an End Date/Time value.

The Start Date/Time format and End Date/Time format depend on the database being used in the Payment Server. Please refer to your Payment Provider for the correct format used by the Payment Server.

**Note:** The capacity in Payment Client 3.1 to handle large results has been upgraded. However, there is a limitation in the Active Server Pages (ASP) environment on the length of a field that can be handled. Therefore when you receive an especially large result from a Reconciliation Query, the ASP environment will truncate the DR when it hits the length limitation. So although the COM interface supports large results, you will not get them if you are using ASP web pages on IIS. However other programming environments that interface to COM (for example VBA and C++) will be able to utilise the large results capacity.

## AMA limited shopping transaction history

A limited shopping transaction history returns some of the OrderIDs reference transactions (shopping transactions) between a start date and an end date where the cardholder entered their card details for a 2-Party or 3-Party transaction. In Purchase mode it would be the Purchase transaction or the Auth transaction in Auth/Capture mode. It also contains some summary information for the entire shopping experience (i.e. including captures, refunds).

The Limited Shopping Transaction History command allows a merchant to limit the number of shopping transactions for the MerchantId in the Payment Server over a specified period.

- The period is specified by the use of a Start Date/Time value and an End Date/Time value.
- The limit is specified by the field '*MaxTrans*'.
- The starting point to start returning transactions from is specified by the field: '*LastTxnReturned*'.

The Start Date/Time format and End Date/Time format depend on the database being used in the Payment Server. Please refer to your Payment Provider for the correct format being used by the Payment Server.

**Note:** The capacity in Payment Client 3.1 to handle large results has been upgraded. However, there is a limitation in the Active Server Pages (ASP) environment on the length of a field that can be handled. Therefore when you receive an especially large result from a Reconciliation Query, the ASP environment will truncate the DR when it hits the length limitation. So although the COM interface supports large results, you will not get them if you are using ASP web pages on IIS. However other programming environments that interface to COM (for example VBA and C++) will be able to utilise the large results capacity.

## AMA financial transaction history

A financial transaction history refers to the series of transactions that have been performed over a time period relating back to a specified Payment Server OrderID (shopping transaction) number.

Input the reference transaction number and the output displays all the results about the transaction and other transactions linked to it.

The number of financial transactions generated from a Payment Server OrderID depends on the type of transaction and the actions performed.

In Purchase mode this only includes the original purchase transaction, plus any refunds and voids that have been performed. In Auth/Capture mode this is the original Auth transaction plus any captures, refunds and voids.

## AMA spanned financial transaction history

A spanned financial transaction history refers to the series of transactions that have been performed over the time period specified by a start date and end date.

Input a Start Date/Time value and an End Date/Time value and it returns all the details about **all the transactions that have occurred in that period** for this MerchantId.

In Purchase mode this only includes the original purchase transaction plus any related refunds and voids that have been performed. In Auth/Capture mode this includes the original Auth transactions plus any related captures, and refunds and voids for those Auth transactions.

The Start Date/Time format and End Date/Time format depend on the database being used in the Payment Server. Please refer to your Payment Provider for the correct format being used by the Payment Server.

**Note:** The capacity in Payment Client 3.1 to handle large results has been upgraded. However, there is a limitation in the Active Server Pages (ASP) environment on the length of a field that can be handled. Therefore when you receive an especially large result from a Reconciliation Query, the ASP environment will truncate the DR when it hits the length limitation. So although the COM interface supports large results, you will not get them if you are using ASP web pages on IIS. However other programming environments that interface to COM (for example VBA and C++) will be able to utilise the large results capacity.

## troubleshooting and FAQs

### troubleshooting

This section contains suggestions and solutions to problems that may occur with your integration.

#### what do we do if a session timeout occurs?

It is possible that while a cardholder is entering their card details at the Payment Server, the session is broken (say a communication failure due to a modem connection dropping off). If this occurs, a cardholder will lose their session. Even if they come back to your site, they will have a new session, and their old session will never be completed.

To determine the status of the lost transaction, you will need to perform a QueryDR transaction based on the original *vpc\_MerchTxnRef*.

#### what does a payment authentication status of "A" mean?

An authentication state of "A" indicates that the authentication transaction failed when the Payment Server tried to authenticate itself with the Directory Server.

A possible reason for the failure is that the 3-D Secure (for Verified by Visa™ and MasterCard SecureCode™) merchant ID or password is set incorrectly for a merchant profile in the Payment Server Merchant Manager, that is, the merchant's Verified by Visa username and password (for Visa) or SecureCode username and password (for MasterCard) have not been set correctly by the Merchant Service Organisation (MSO) in the merchant profile.

#### does the cardholder's internet browser need to support cookies?

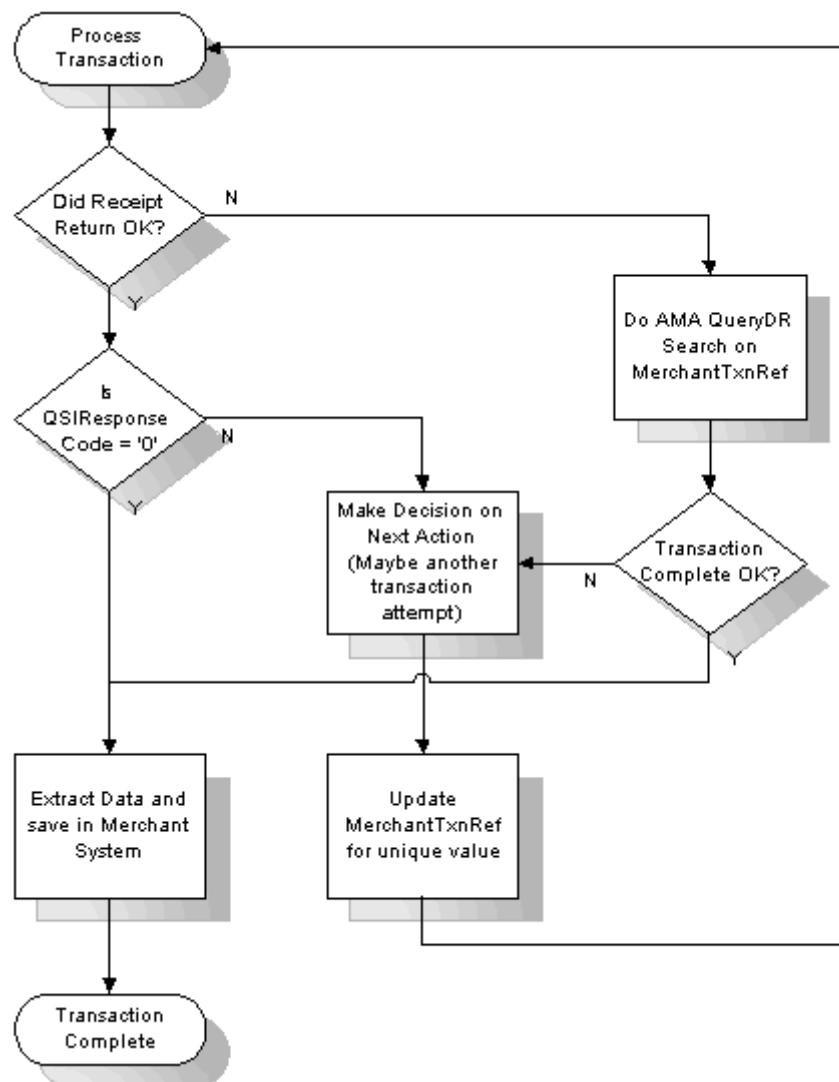
The Virtual Payment Client interface requires a cardholder's browser to support cookies for all 3-Party.

## what happens if a transaction response fails to come back?

The two ways of dealing with a Transaction Response that fails to come back are:

- Flag the transaction as having an error that the merchant needs to manually check using Merchant Administration on the Payment Server.
- Utilise Advanced Merchant Administration (AMA) commands to search the Payment Server database for the transaction by using the *QueryDR* command. The *vpc\_MerchTxnRef* is used as the transaction identifier when searching using *QueryDR*.

Because the Transaction Response has failed to come back, there is no transaction number available from the Payment Server to identify the transaction in question, and this is why you use the *vpc\_MerchTxnRef*. It is important to have a unique *vpc\_MerchTxnRef* for every transaction otherwise the query could return multiple results. Only the most recent transaction is returned in the *QueryDR* command if there are multiple results, but this may not be the transaction you are concerned with.





When you find the required *vpc\_MerchTxnRef* in the *QueryDR*, check if it is successful by the *vpc\_TxnResponseCode* field (equal to '0'). If the *vpc\_TxnResponseCode* is zero, then the transaction is successful and you just need to extract the relevant data details from the *QueryDR* results for your records. If the *vpc\_TxnResponseCode* is not 0, you need to determine the next course of action based on what you would do if the *vpc\_TxnResponseCode* were not 0 in a normal Transaction Response coming back from the Payment Server.

If you query the Payment Server for the *vpc\_MerchTxnRef* using the *QueryDR* call and you do not receive any results (*vpc\_DRExists* = 'N'), then it is safe to repeat the transaction. It is safe to use the same *vpc\_MerchTxnRef*, as the existing one does not show up in the Payment Server's database and was therefore never processed.

If the *QueryDR* is flagged as having multiple results (returns 'Y' in the *vpc\_FoundMultipleDRs* field), the *vpc\_MerchTxnRef* is not unique and you will have to manually check all the results for the same time when the *vpc\_MerchTxnRef* number was created. This is one of the primary reasons for implementing a unique *vpc\_MerchTxnRef* for every transaction.

## frequently asked questions

### what is an outage?

An outage is considered a “production fault” as it means that the Payment Server is temporarily offline; for example for maintenance and upgrades, and so forth.

During an outage, all transactions are declined with an error message indicating that the service is currently unavailable.

### how do i know if a transaction has been approved?

All approved transactions are represented with a `vpc_TxnResponseCode` of '0' (zero) from the Payment Server. Any other code represents a declined or failed transaction.

### can the payment server's payment pages be modified for a merchant?

No. The Payment Server's payment pages are branded using either the Payment Provider's or Bank's branding to assure cardholders of the security of the transaction. If you do not wish to display the Payment Provider's branded pages you need to implement either the 3-Party with card details, or the 2-Party integration models.

**Note:** Using the 2-Party Integration model prohibits the use of 3-D Secure Verified by Visa™ and MasterCard SecureCode™ functionality.

### is a shopping cart required?

It is not necessary to have a shopping cart. All that is required is that the transaction information is within the Transaction Request passed to the Payment Server.

### what is merchant administration?

Merchant Administration allows merchants to use an Internet browser to monitor and manage electronic transactions through a series of easy to use pages. It allows merchants to interactively perform historical searches, captures, refunds and setup activities.

To use Merchant Administration, you need access to the Internet through a browser (such as Internet Explorer or Netscape), your Payment Provider's URL (or web site address) and a merchant profile. The merchant profile is a record of your details and privileges, which is stored on the Payment Server. For more details, please refer to the *Merchant Administration User Guide*.

## how much will it cost to keep the payment site running?

The Virtual Payment Client is very stable, is not difficult to keep running and requires no more maintenance than the web server itself.

## does the payment server handle large peaks in transaction volumes?

The Payment Server queues pending transactions so transactions are not lost, although the cardholder may at times notice a slight delay when transactional loads are extremely high.

## how long will an authorisation be valid on a cardholder account?

This depends on the Financial Institution who issued the card to the cardholder. Each card Issuer defines the authorisation expiry period in which they hold the funds on the cardholder's account, while they wait for the arrival of the capture transaction. Generally it is 5-8 processing days, before the authorisation purges from the cardholder account and access to the funds are released back to the cardholder.

## what is the RRN and how do I use it?

RRN (Reference Retrieval Number) is a unique number for a particular MerchantId. This is the value that is passed back to the cardholder for their records. You cannot search for this field in Merchant Administration, but it is displayed in Merchant Administration on the transaction details pages as the Reference Retrieval Number (RRN). It is one of the fields returned in a *QueryDR* and the transaction result (captures, refunds).

The RRN is useful when your application does not provide a receipt number. The RRN can be viewed in Merchant Administration.

## what is the difference between RRN, MerchTxnRef, OrderInfo, Authorizeld and TransNo?

- *RRN* (Reference Retrieval Number) is a unique number assigned to each transaction for a particular MerchantId. This is the value that is passed back to the cardholder for their records. You cannot search on this field in Merchant Administration, but it is displayed in Merchant Administration on the transaction details pages as the Reference Retrieval Number (RRN). It is one of the fields returned in a queryDR and the transaction result (captures, refunds).
- *MerchTxnRef* is generated by your merchant application. Ideally it should be a unique value for each transaction and you should retain this number so that transactions can be searched for in your application and the Payment Server. See Merchant Transaction Reference (vpc\_MerchTxnRef).
- *OrderInfo* is also generated by your application. It should also be a unique value for each order, which you should retain so that you can search for the transaction in your application and the Payment Server.
- *Authorizeld* is an identifier from the Acquiring Bank, which is in the Transaction Response for the authorisation. This field cannot be searched for in Merchant Administration, but it is displayed in Merchant Administration as the Authorisation Code. It is one of the fields returned in a transaction result and an AMA QueryDR.
- *TransNo* or *TransactionNo* is a unique number for each MerchantId generated by the Payment Server that is called the OrderID or shopping transaction number. The OrderID is the key reference value for transactions when using AMA transactional functions like captures and refunds.

## advanced function compatibility

The following table lists the common functions available on the Payment Server and the compatibility of functions the merchant can use. To determine the functionality that can be included in a Digital Order, choose a function in a column and follow it down to the appropriate row.

✓ Enabled for this transaction type or compatible with this feature.

✗ Not enabled for this transaction type or not compatible with this feature

Supplementary Feature Compatibility	Address Verification	Card Security Code	Ticket Number	External Payment Selection	Card Details in 3-Party	3-D Secure Authentication & Payment	Card Present	CPC 2
2-Party Transaction	✓	✓	✓	✗	✗	✗	✓	✓
3-Party Transaction	✓	✓	✓	✓	✓	✓	✗	✓
Address Verification		✓	✓	✓	✓	✓	✓	✓
Card Security Code	✓		✓	✓	✓	✓	✓	✓
Ticket Number	✓	✓		✓	✓	✓	✓	✓
External Payment Selection	✓	✓	✓		✓	✓	✗	✓
Card Details in 3-Party	✓	✓	✓	✓		✓	✗	✓
3-D Secure Authentication & Payment	✓	✓	✓	✓	✓		✗	✓
Card Present	✓	✓	✓	✗	✗	✗		✓

## suggested merchant actions

Acq Resp Code	Recur Resp Code	Merchant Advice Description	Examples of reason for decline	Suggested Merchant Action
DE39	DE48 SE84			
00 05 14 51 54	01	New account information available	<ul style="list-style-type: none"> <li>Expired card</li> <li>Account upgrade</li> <li>Portfolio sale</li> <li>Conversion</li> </ul>	Obtain new account information before next billing cycle.
51	02	Try again later	<ul style="list-style-type: none"> <li>Over credit limit</li> <li>Insufficient funds</li> </ul>	Recycle transaction 72 hours later.
05 14 51 54	03	Do not try again	<ul style="list-style-type: none"> <li>Account closed</li> <li>Fraudulent</li> </ul>	Obtain another type of payment from customer.

# index

## 2

- 2-Party Payments Information Flow • 28
- 2-Party Payments Integration Model • 12

## 3

- 3-Party Payment Information Flow • 30
- 3-Party Payments Integration Model • 11
- 3-Party Payments using Verified by Visa™ and MasterCard SecureCode™ • 39
- 3-Party Payments where the merchant collects all the cardholder's card details • 38
- 3-Party Payments where the merchant collects the cardholder's card type • 38

## A

- About e-Payment Information Flows • 8
- About this Document • 2
- Additional 3 Party Functionality • 38
- Additional Features for 3-Party Transactions • 25
- Address Verification Service (AVS) • 40
- Advanced Function Compatibility • 40, 71
- Advanced Merchant Administration (AMA) • 3, 58
- Advantages and Disadvantages of the 3-D Secure modes of transaction • 52
- Airline Ticket Number • 44
- AMA Financial Transaction History • 63
- AMA Limited Shopping Transaction History • 63
- AMA Shopping Transaction History • 62
- AMA Spanned Financial Transaction History • 64
- Audience • 1
- Authorisation in the Auth/Capture Model • 9
- Authorisation/Capture Model • 9, 15
- Automatically Check the Integrity of 3 Party Transactions Using Secure Hash • 21

## B

- Bank Account Type • 47
- Best Practices to Ensure Transaction Integrity • 23

## C

- Can the Payment Server's Payment Pages be Modified for a Merchant? • 68
- Capture • 58
- Capture in the Auth/Capture Model • 10
- Card Holder Name Transactions • 40

- Card Present Transactions • 42
- Card Security Code (CSC/CVV2) • 41
- Check for a replay of a transaction • 24
- Check for suspect transactions • 24
- Check That the Field Values in the Response Match Those in the Request • 24
- Commence Live Online Payments • 18
- Conduct Final Pre-Production testing • 18
- Creating an MD5 Signature for 3-Party Transactions • 26

## D

- Design and Implement the Integration • 17
- Detect alteration of Requests and Responses using Secure Hash • 25
- Determine Any Advanced Functionality • 15
- Determine the Input and Output Fields • 17
- Determine the Payment Model • 15
- Determine Your Integration Model • 15
- Disclaimer • 2
- Does the Cardholder's Internet Browser Need to Support Cookies? • 65
- Does the Payment Server Handle Large Peaks in Transaction Volumes? • 69

## E

- Ensuring Successful Payments • 20
- Excessive Captures • 62
- External Payment Selection (EPS) • 42

## F

- Frequently Asked Questions • 68

## G

- Go Live • 18

## H

- Handle a Transaction Request • 35
- Handle a Transaction Response • 36
- Handle Session Variables • 17, 37
- How Do I know If a Transaction Has Been Approved? • 68
- How Do My Cardholders Know If My Site is Using SSL? • 22
- How e-Payments Transfer Funds • 7
- How Long Will an Authorisation be Valid on a Cardholder Account? • 69

How Much Will it Cost to Keep the Payment Site Running? • 69

## I

Information Flow of a 3D-Secure Authentication/Payment transaction • 50

Integrating 2-Party Payments • 2, 28, 57

Integrating 3-Party Payments • 2, 30

Integrating 3-Party Payments with Virtual Payment Client • 35

Integration Models and Communication Methods • 11, 15

Introduction • 2

Is a Shopping Cart required? • 68

## L

Look Up Your Access Code and Secure Hash Secret in Merchant Administration • 16

## M

Manually Check Transaction Results Using Merchant Administration • 20

Merchant Transaction Frequency • 43

Merchant Transaction Reference (vpc\_MerchTxnRef) • 19

Merchant Transaction Source • 43

Mode 1 - Implementing a 3 Party Authentication and Payment transaction (Payment Server collects card details) • 53

Mode 2 - Implementing a 3 Party Authentication and Payment transaction (Merchant collects card details) • 54

Mode 3a - Implementing a 3 Party Style Authentication Only transaction • 55

Mode 3b - Implementing a 2-Party Style Pre-Authenticated Payment Transaction • 57

## N

Nominal Auth/Purchase Mode • 10

## O

Obtain an E-commerce Merchant facility • 16

## P

Payment Authentication • 13, 47

Payment Authentication 3-D Secure transaction modes • 48

Payment Models • 8

Perform a Basic Test Transaction Using the Supplied Example Code • 17

Pre-Authorisation/Purchase Mode • 9

Preface • 1

Preparing for Integration • 2, 8, 11

Prerequisites • 15

Protecting Cardholder Information Using SSL • 22

Provide Your Financial Institution Merchant Number, Terminal Id/s and MCC to your Payment Provider • 16

Purchase Model • 8

## Q

QueryDR • 61

## R

Reference Fields • 19

Referral Message • 44

Referral Transaction • 44

Refund • 59

Related Documents and Materials • 3

Risk Management • 45

## S

Securing Your Payments • 2, 15, 20, 22

Selection Guidelines for Integration Models • 13

Sending Session Variables to the Payment Server • 37

Standalone Capture • 59

Standalone Refund • 59

Store Secure Hash Secret Securely • 25, 27

Suggested Merchant Actions • 72

Supplementary Transactions • 2, 40

Support Material and Information • 15

## T

Terminology • 4

Test Your Integration • 18

The Components of an e-Payment Solution • 7

The Merchant Application • 8

The Virtual Payment Client • 8

Troubleshooting • 65

Troubleshooting and FAQs • 3, 65

## U

Understanding e-Payments • 2, 6

Use a Unique MerchTxnRef for Each Transaction Attempt • 23

Use Good Password Security for Merchant Administration • 24

Using 3-D Secure Payment Authentications • 23, 38, 39

## V

Validate the SSL Certificate of the Payment Server • 24



Virtual Payment Client Integration Guidelines • 19  
Virtual Payment Client Order Information (vpc\_OrderInfo) •  
20  
Void Capture • 59  
Void Purchase • 61  
Void Refund • 60

## W

What are e-Payments? • 6  
What Do We Do if a Session Timeout Occurs? • 65  
What Does a Payment Authentication Status of • 65  
What happens if a Transaction Response fails to come  
back? • 66  
What is an Outage? • 68  
What is Merchant Administration? • 68  
What is the Difference Between RRN, MerchTxnRef,  
OrderInfo, Authorizeld and TransNo ? • 70  
What is the RRN and How Do I Use It? • 69  
What the Cardholder Sees • 29, 31  
What the Payment Server does • 36  
When to combine 3-Party and 2-Party Payments • 14  
When to use 2-Party Payments • 13  
When to use 3-Party Payments • 13  
Where to Get Help • 1