Virtual Payment Client

Integration Guide

Version 4.2.0

For TNSPay 4.2

Disclaimer

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

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

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

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

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

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

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

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

Contents

CHAPTER 1

Preface

Welcome to TNS

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

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

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

Audience

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

Where to Get Help

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

CHAPTER 2

Introduction

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

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

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

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
Understanding e-Payments on page 13	Describes how e-Payments work**.**
Preparing for Integration on page 19	Describes the various options and models you need to choose before commencing your integration.
Securing your Payments on page 31	Describes the security features available for the Virtual Payment Client
2-Party Payments on page 37	Describes the information flow and integration model for 2-Party Payments.
3-Party Payments on page 39	Describes the information flow and integration model for 3-Party Payments.
Supplementary Transactions on page 53	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.
Advanced Merchant Administration (AMA) Transactions on page 77	Describes the flows for each AMA transaction/query, and the data requirements for each stage of the transaction/query.
Troubleshooting and FAQs on page 85	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 by TNS to illustrate the use of the Virtual Payment Client API.

Terminology

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

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

CHAPTER 3

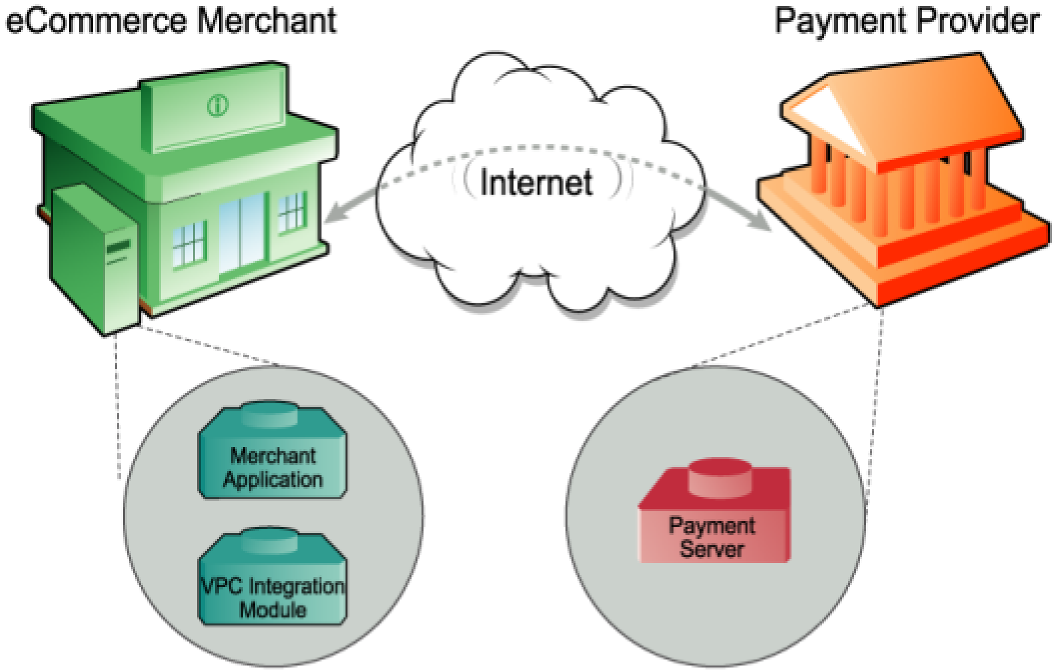
Understanding e-Payments

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; and
- Sends the information to the Payment Server
- The Virtual Payment Client 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*** on page 19.

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.

CHAPTER 4

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 3-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 3-D Secure. 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 a-synchronous 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 the 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 3-D Secure.

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.

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, 3-D Secure).
- 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, 3-D Secure). For more ***information see Payment Authentication*** on page 66.
- 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, or
- 2-Party Payments Integration Model.
- Combination of 2-Party and 3-Party Integration

For more information, please refer to ***Integration Models*** on page 19.

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*** on page 15.

Determine Any Advanced Functionality

The available advanced functionality includes:

- 3-D Secure. See ***Securing your Payments*** on page 31.
- 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**
  The Secure Hash Secret is a key used as the initial piece of encryption data to create a SHA256 HMAC. 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 TNS example code before you implement the integration with your application:

- Validates that your system is set-up correctly and
- 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 a 3-Party integration (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*** on page 45.

  Session variables are not required when using the 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.

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.

CHAPTER 5

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 value is displayed with the transaction in Merchant Administration, and can also be used in transaction search criteria.

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.

Merchant Order Reference (vpc_OrderInfo)

vpc_OrderInfo is an identifier provided by you to identify the order on the Payment Server database. This value will be displayed in the Merchant Administration portal when manually searching for orders. It can be an order number, an invoice number, or a shopping cart number. The vpc_OrderInfo field should be the same for each transaction against the order 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 an order 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 ensuring successful payments for the goods shipped. This includes ensuring 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 3-D Secure. See ***Securing your Payments*** on page 31.
- 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 involved in manual checks is the possibility of incorrectly matching OrderIDs with the cardholder's orders. Also as volumes grow, the risk may become significant as would the time and cost involved in completing the task.

Ensure Correct Character Encoding

The TNS Payment Server only allows the use of ISO 8859-1 (Latin 1) characters. By default all incoming data is assumed to be encoded as ISO 8859-1. This default encoding is also safe for incoming data encoded as US-ASCII.

Note: The Payment Server will still only accept characters which are valid in the ISO 8859-1 encoding. Providing data in an encoding such as UTF-8 will ensure that the Payment Server correctly interprets those characters, but it will still reject the request if any character cannot be represented in ISO 8859-1.

2-Party Payment Model

If the incoming data is not encoded as ISO 8859-1 or US-ASCII, the content-type header must be set on the HTTP Post message with the correct encoding. An example content type header would be:

“Content-Type: application/x-www-form-urlencoded; charset=UTF-8"

The Payment Server would then decode incoming data as UTF-8.

3-Party Payment Model

If you are using HTTP POST to pass in data which is not encoded as ISO 8859-1 or US-ASCII, then you must set the accept-charset form tag to ISO-8859-1 on the post message. This instructs the browser to convert the data to a format that can be correctly decoded by the Payment Server.

If you are using HTTP GET to pass in data then you **MUST NOT** pass in any characters outside of the ISO 8859-1 character set. Any such characters cannot be correctly decoded by the Payment Server.

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.

CHAPTER 6

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.

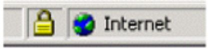
When implementing the 2-Party or 3-Party Integration models you must ensure your application presents a secure form using SSL. You should also consider using a secure form in your application when collecting confidential information such as cardholder addresses.

Note: When implementing 3-Party integrations, the merchant's website must enforce SSL communications to avoid 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 3-D Secure, 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 3-D Secure, 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.

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, 498765XXXXXXX769).

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.

Detect alteration of Requests and Responses using Secure Hash

The Virtual Payment Client uses SHA-256 HMAC Secure Hash, which plays a role in transaction security as it is 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 a SHA-256 HMAC Secure Hash, which is appended to the Transaction Request. Since 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 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.

Also, it is strongly recommended that you generate the Secure Hash Secret using SHA-256 HMAC for all the new merchant integrations.

For more information on Secure Hash Secret, see ***Store Secure Hash Secret securely*** on page 35.

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.

CHAPTER 7

Integrating 2-Party Payments

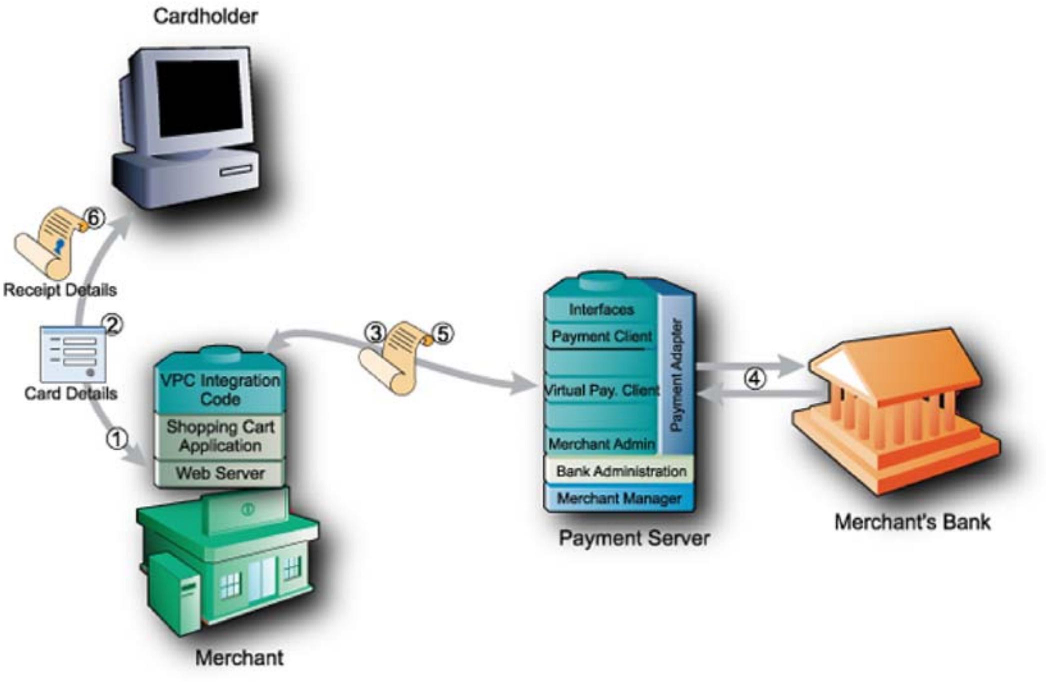
In the 2-Party Integration Model, a cardholder places an order and provides their card details (card type, card number and expiry date) to you by **M**ail **O**rder or by **T**elephone **O**rder (**MOTO** transactions) including Interactive Voice Response (IVR) systems, or some card present application 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 delegating these duties to the Payment Server by using a 3-Party integration.

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 cardholders 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

With 2-Party Payments over the Internet, the merchant integration is responsible for capturing the cardholder details and presenting them with a receipt after the transaction has been processed by the Payment Provider.

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.

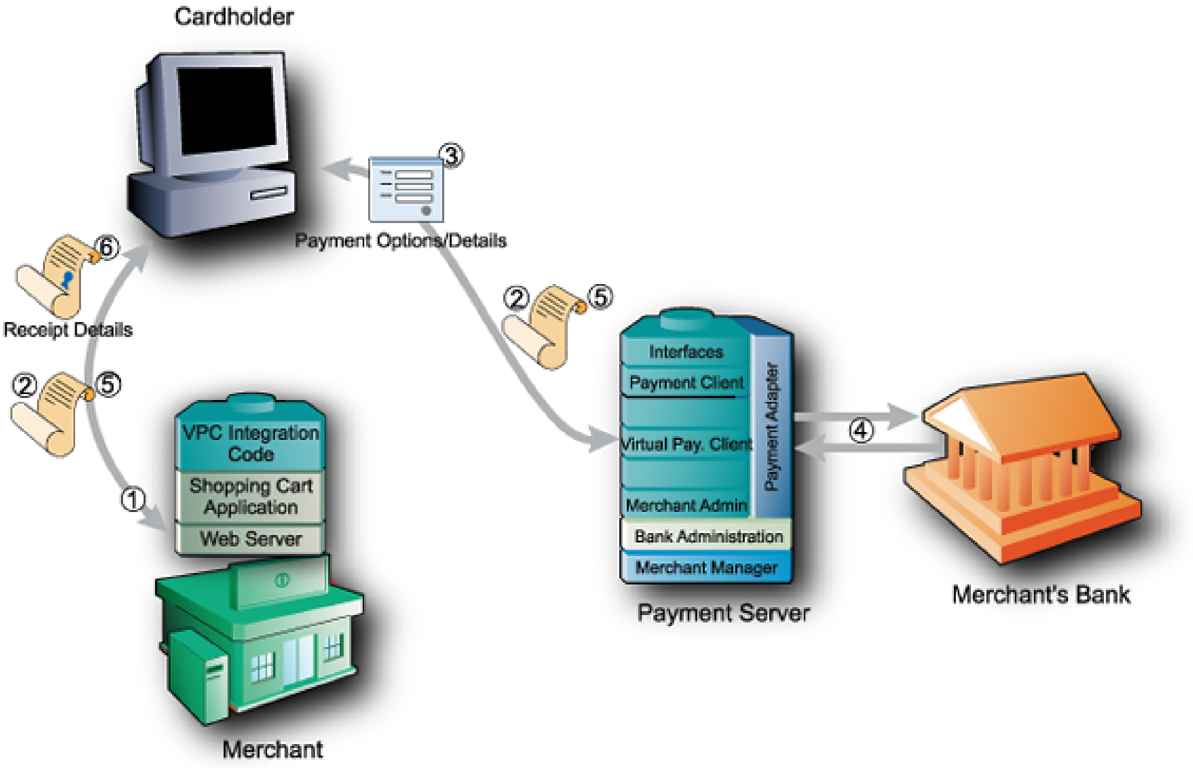
CHAPTER 8

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.

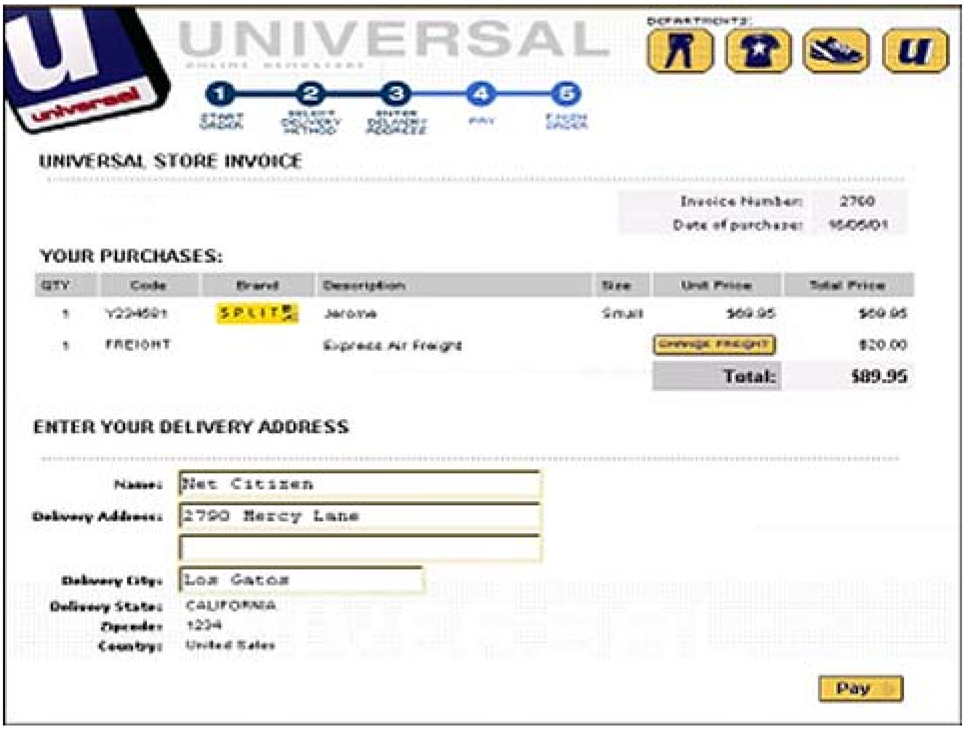
5 The Payment Server then redirects the cardholder back to merchant's site with the transaction response. The response contains the result of the transaction.

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



UNIVERSAL STORE INVOICE

Invoice Number: 2760
Date of purchase: 16/06/01

YOUR PURCHASES:

QTY	Code	Brand	Description	Size	Unit Price	Total Price
1	Y224501	SPLIT	Jerome	Small	$69.95	$69.95
1	FREIGHT		Express Air Freight		CHANGE FREIGHT	$20.00
					Total:	$89.95

ENTER YOUR DELIVERY ADDRESS

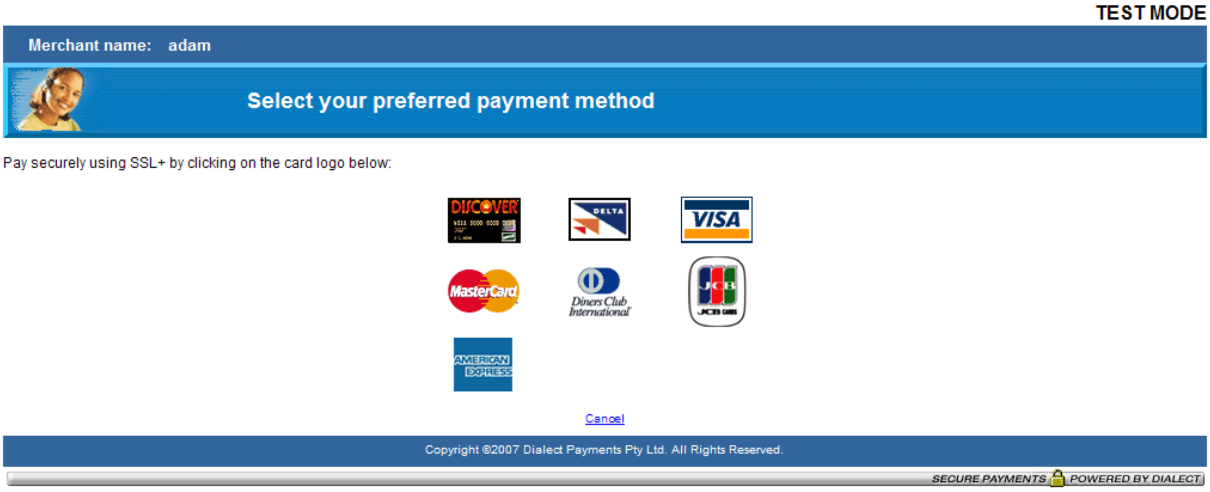
Name: Net Citizen
Delivery Address: 2790 Mercy Lane
Delivery City: Los Gatos
Delivery State: CALIFORNIA
Zipcode: 1234
Country: United Sates

Pay

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.



TEST MODE

Merchant name: adam

Select your preferred payment method

Pay securely using SSL+ by clicking on the card logo below:

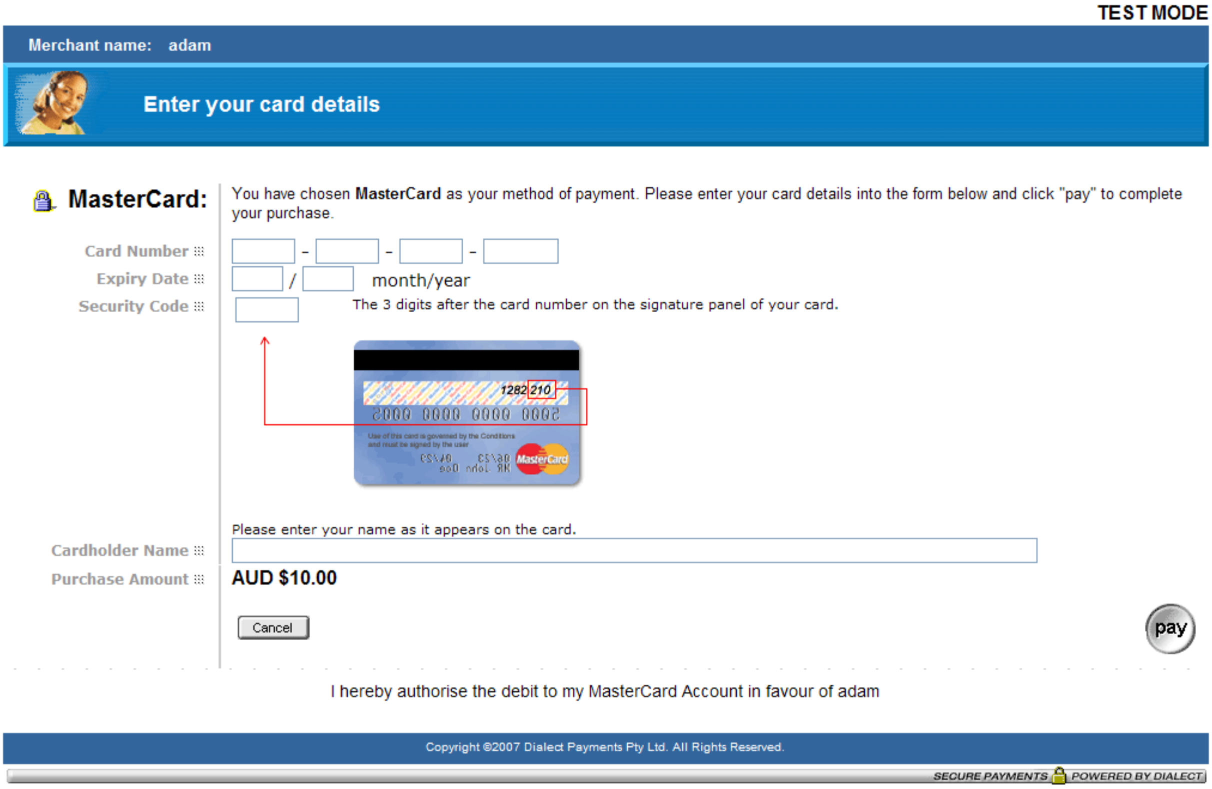
Cancel

Copyright ©2007 Dialect Payments Pty Ltd. All Rights Reserved.

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



TEST MODE

Merchant name: adam

Enter your card details

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.

Card Number ::: ____ - ____ - ____ - ____

Expiry Date ::: ____ / ____ month/year

Security Code ::: ____ The 3 digits after the card number on the signature panel of your card.

Please enter your name as it appears on the card.

Cardholder Name ::: ____

Purchase Amount ::: **AUD $10.00**

Cancel

pay

I hereby authorise the debit to my MasterCard Account in favour of adam

Copyright ©2007 Dialect Payments Pty Ltd. All Rights Reserved.

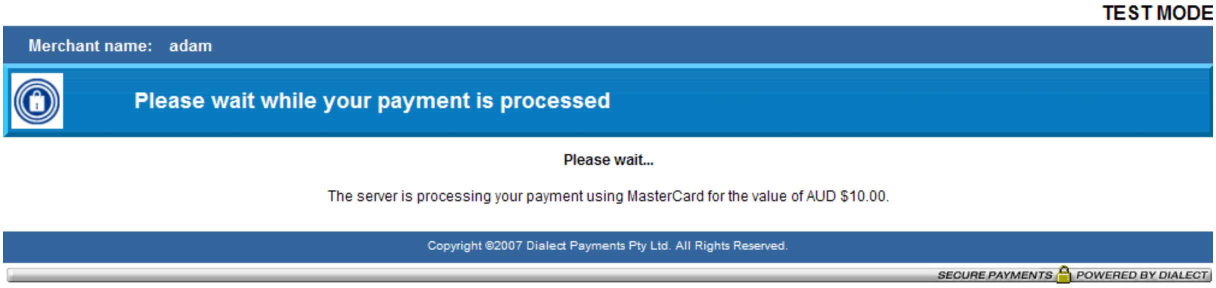
SECURE PAYMENTS POWERED BY DIALECT

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

Please wait while your payment is processed

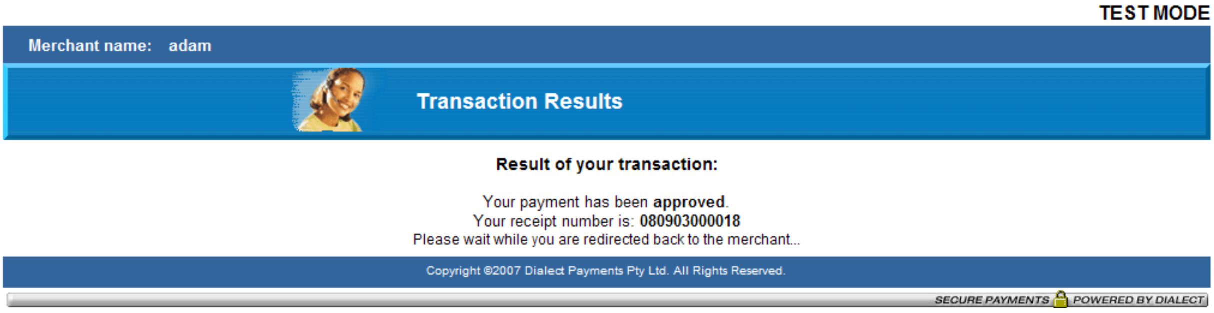
Please wait...

The server is processing your payment using MasterCard for the value of AUD $10.00.

Copyright ©2007 Dialect Payments Pty Ltd. All Rights Reserved.

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

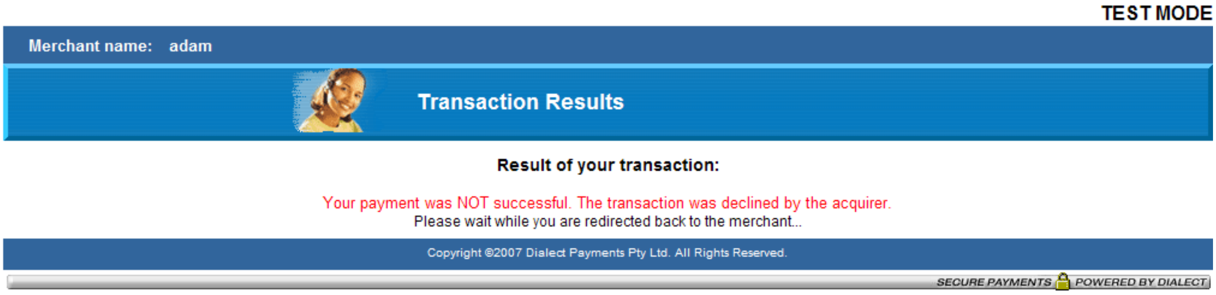
Transaction Results

Result of your transaction:

Your payment has been **approved**.
Your receipt number is: **080903000018**
Please wait while you are redirected back to the merchant...

Copyright ©2007 Dialect Payments Pty Ltd. All Rights Reserved.

or, if declined, the following page would appear:



TEST MODE

Merchant name: adam

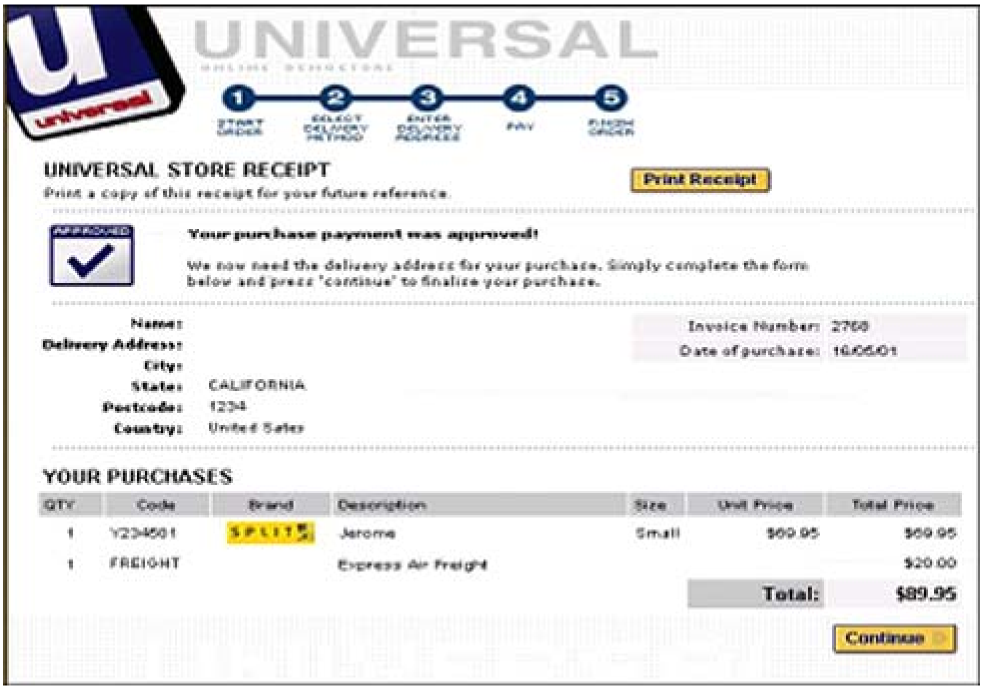
Transaction Results

Result of your transaction:

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.

6 The merchants application receipt page.



UNIVERSAL STORE RECEIPT

Print a copy of this receipt for your future reference.

Your purchase payment was approved!

We now need the delivery address for your purchase. Simply complete the form below and press 'continue' to finalise your purchase.

Name:
Delivery Address:
City:
State: CALIFORNIA
Postcode: 1234
Country: United Sates

Invoice Number: 2760
Date of purchase: 16/05/01

YOUR PURCHASES

QTY	Code	Brand	Description	Size	Unit Price	Total Price
1	Y234561	SPLITS	Jerome	Small	$69.95	$69.95
1	FREIGHT		Express Air Freight			$20.00
					Total:	$89.95

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 SHA256 HMAC
- Send it with the Transaction Request, and
- Check the SHA256 HMAC in the transaction 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 SHA256 HMAC.

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%5FCom
mand=pay&vpc%5FAccessCode=A8698556&vpc%5FMerchTxnRef=123&vpc%5FMerchant=TESTMERCHAN
T&vpc%5FOrderInfo=Example&vpc%5FAmount=100&vpc%5FReturnURL=http%3A%2F%2FMerchant_We
b_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 SHA256 HMAC is incorrect, the Payment Server:
  - Returns the cardholder back to the merchant with a Transaction Response with an error message indicating the SHA256 HMAC 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_AVSResultCode=Unsupported&vpc_AcqAVSRespCo
de=Unsupported&vpc_AcqCSCRespCode=Unsupported&vpc_AcqResponseCode=00&vpc_Amount=100
&vpc_AuthorizeId=020072&vpc_BatchNo=20051209&vpc_CSCResultCode=Unsupported&vpc_Card
=MC&vp
```

The merchant application receipting function needs to be able to calculate the SHA256 HMAC in the Transaction Response to determine if the signature received is valid for the receipt data. It has to handle:

- Incorrect SHA256 HMAC
- 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. With SHA-256 HMAC Secure Hash, you must not use session variables. For more information, see *Creating a SHA-256 HMAC Secure Hash* in *Virtual Payment Client Reference Guide*.

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 3-D Secure. For more information, see ***Using 3-D Secure Authentication*** on page 32.

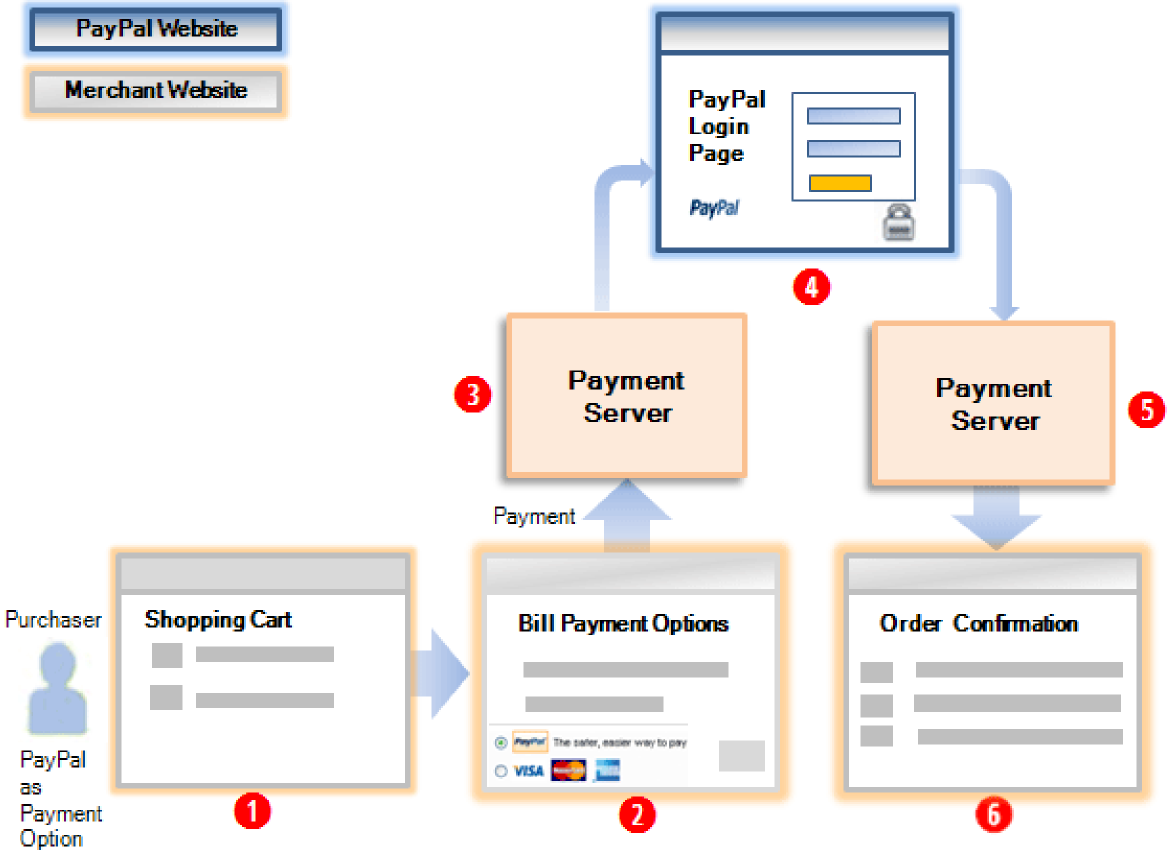
3-Party Payments where the merchant collects the cardholder's card type

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 using PayPal

In 3-Party Payments you can have your Payment Provider enable you to use PayPal, which provides additional payment options to your customers. This type of transaction requires the 3-Party model, and works by redirecting the purchaser to the PayPal website where the purchaser completes the transaction. No Payment Server pages are displayed for card selection or card details entry.



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

❶	A purchaser browses your online store, selects a product and may enter their shipping details into the online store at the checkout page.
❷	The purchaser clicks **PayPal** button and your online store sends the payment request to the Payment Server through an API call.
❸	The Payment Server redirects the purchaser to the PayPal website.
❹	The purchaser logs into their PayPal account and selects a funding source and approves the request for payment. Depending on the request made by the merchant, PayPal may request a shipping address from the purchaser. The purchaser may select a shipping address previously stored with PayPal, or enter a new one.
❺	The purchaser is redirected from PayPal back to the merchant’s website, via the Payment Server, with the transaction response. The response contains the result of the transaction.
❻	The merchant's website interprets the response and displays the receipt and confirms the order to the purchaser for their records.

Checkout from a Shopping Cart

To integrate with PayPal, the merchant's website must use either of the following methods to enable the purchaser to checkout from the shopping cart.

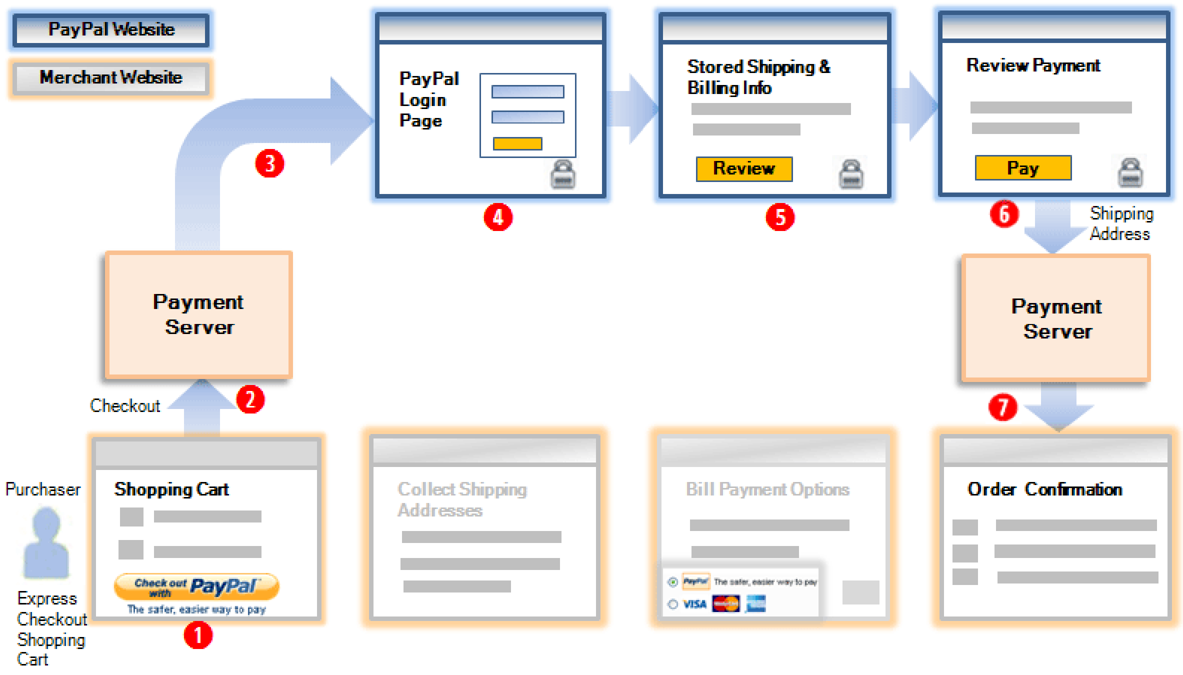
- PayPal displayed as a checkout choice on the shopping cart screen, usually represented with Checkout with PayPal logo.

  This method of payment is called Shortcut. For details, see ***Shortcut Payment Flow*** on page 49.
- PayPal displayed along with other credit cards as a payment method on the billing screen, usually represented with the Paypal logo

  This method of payment is called Mark. For details, see ***Mark Payment Flow*** on page 50.

Shortcut Payment Flow

The Shortcut method redirects the purchaser from the shopping cart of the merchant's website to the PayPal website by bypassing the payment option page and the shipping address details page of the merchant's website. PayPal collects the shipping address details on behalf of the merchant.



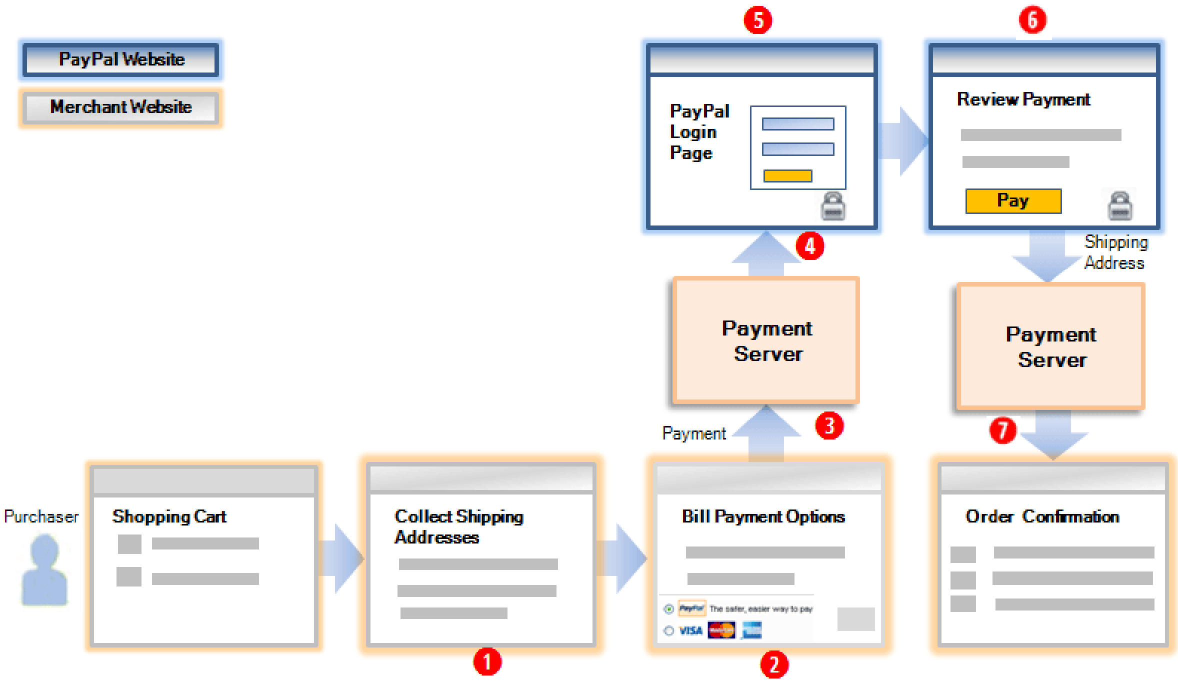
The payment flow is as follows:

❶	A purchaser browses your online store, selects a product and selects **Checkout with PayPal** in the online store at the checkout page.
❷	The merchant performs a 3-Party API call to the Payment Server when the purchaser clicks **Checkout with PayPal**.
❸	The Payment Server redirects the purchaser to the PayPal website.
❹	The purchaser logs into their PayPal account using the PayPal Login page.
❺	The purchaser specifies the shipping address. The purchaser may select a shipping address previously stored with PayPal, or enter a new one.
❻	The purchaser selects a funding source and approves the request for payment.
❼	The purchaser is redirected from PayPal back to the merchant's website, via the Payment Server, where an order confirmation screen is displayed to the purchaser.

Note: 3-Party payment pages are **NOT** displayed in this payment flow. The purchaser is present throughout the transaction cycle.

Mark Payment Flow

The Mark method allows the purchaser to enter the shipping address details and select the payment method (PayPal) in the shopping cart of the merchant's website and then redirects the purchaser to the PayPal website.



The payment flow is as follows:

1	A purchaser browses your online store, selects a product and enters the shipping details into the online store at the checkout pages.
2	The purchaser selects **PayPal** from the merchant's payment options screen.
3	The merchant performs a 3-Party API call to the Payment Server when the purchaser selects **PayPal** as the payment option.
4	The Payment Server redirects the purchaser to the PayPal website.
5	The purchaser logs into their PayPal account using the PayPal Login page.
6	The purchaser selects a funding source and approves the request for payment.
7	The purchaser is redirected from PayPal back to the merchant's website, via the Payment Server, where an order confirmation screen is displayed to the purchaser.

Note: 3-Party payment pages are **NOT** displayed in this payment flow. The purchaser is present throughout the transaction cycle.

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 3-D Secure

In 3-Party Payments you can have your Payment Provider enable you to use 3-D Secure, 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*** on page 32.

3-D Secure 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 3-D Secure 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.

CHAPTER 9

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*** on page 90.

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: Not all issuers and acquirers support AVS, so even if AVS data is passed in the transaction data, it may be ignored.

Advanced Address Verification (AAV)

Advanced Address Verification (AAV) is very similar to Address Verification Service (AVS), but containing much more data. AVS is a subset of AAV data and is a security feature used for card not present transactions that compares the cardholder's card billing address entered by the cardholder with the records held in the card issuer's database, plus more information about the transaction.

Once the transaction is successfully processed and authorized, the card issuer returns an address verification result code (AVS result code) in its authorisation response message verifying the level of accuracy required to match the address.

The cardholder's card billing details are sent with the Transaction Request and 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 is successful or will fail.

The Payment Provider has a default match level set in the Payment Server. Anything with an AAV match equal to or higher than the match level in the Payment Server will allow the transaction to be completed. Any data match that is less than the default level results in the transaction failing.

AAV data is suitable for 2-Party transactions where the data must be provided in the Transaction Request. In a 3-Party transaction the payload can be a bit too large to perform a browser redirect; however, 3-Party transactions do support AVS data and its extended fields (BillTo_Firstname and BillTo_Lastname). Only the address details supplied by the cardholder for verification are returned in the Transaction Response.

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

The AVS fields of AAV data is a mandatory transaction requirement in some countries or regions. Please check with your Payment Provider to determine the legal requirements of your region.

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

Airline Passenger Data

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

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

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

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.

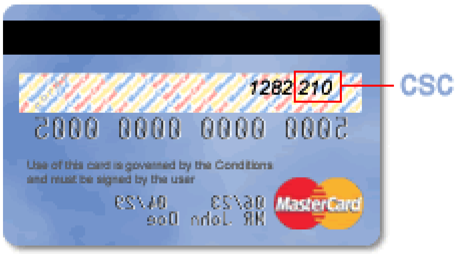
Note: Applies only to 3-party transactions.

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/CVV2)

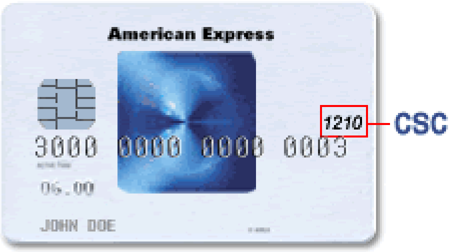
The Card Security Code (CSC) is a security feature used for card not present transactions that compares the Card Security Code on the card with the records held in the card issuer's database.

For example, on Visa and MasterCard credit cards, it is the three-digit value printed on the signature panel on the back.



1282 210
CSC
VISA

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

  or

- 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.

Note: PayPal transactions are processed through 3-Party using EPS, where PayPal payment option is selected on the merchant's application.

Internet Transaction Data (ITD)

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

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

Manual Authorisation ID

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

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** on page 59'.

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

Referral Transaction

Referral transactions allows 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*** on page 59.

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, AVS, CSC — each rule contributes differently to the risk profile. IP Address Range and Card BIN rules enable blocking/reviewing 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/review/accept transactions based on authentication states and IP Country rules enable you to block/review countries with high-risk IP addresses. AVS/CSC rules allow you to block/review/accept transactions based on AVS/CSC response codes.

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.

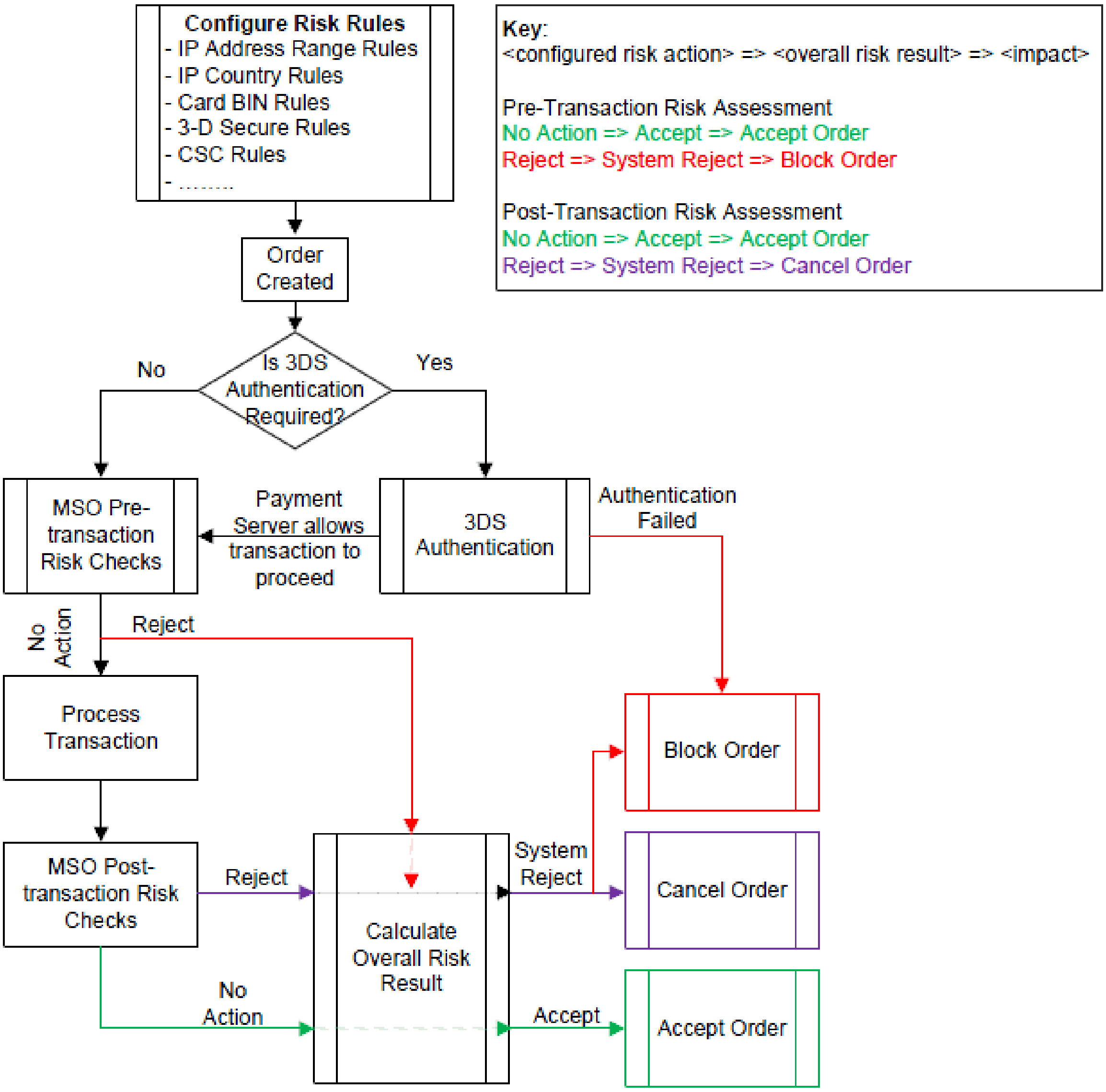
Risk Management is available for both 2-party and 3-party transactions. Though risk rules can be configured only through the Merchant Administration or Merchant Manager portal, transactions processed through the Virtual Payment Client will be assessed for risk, and the overall risk result for each authorisation 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.

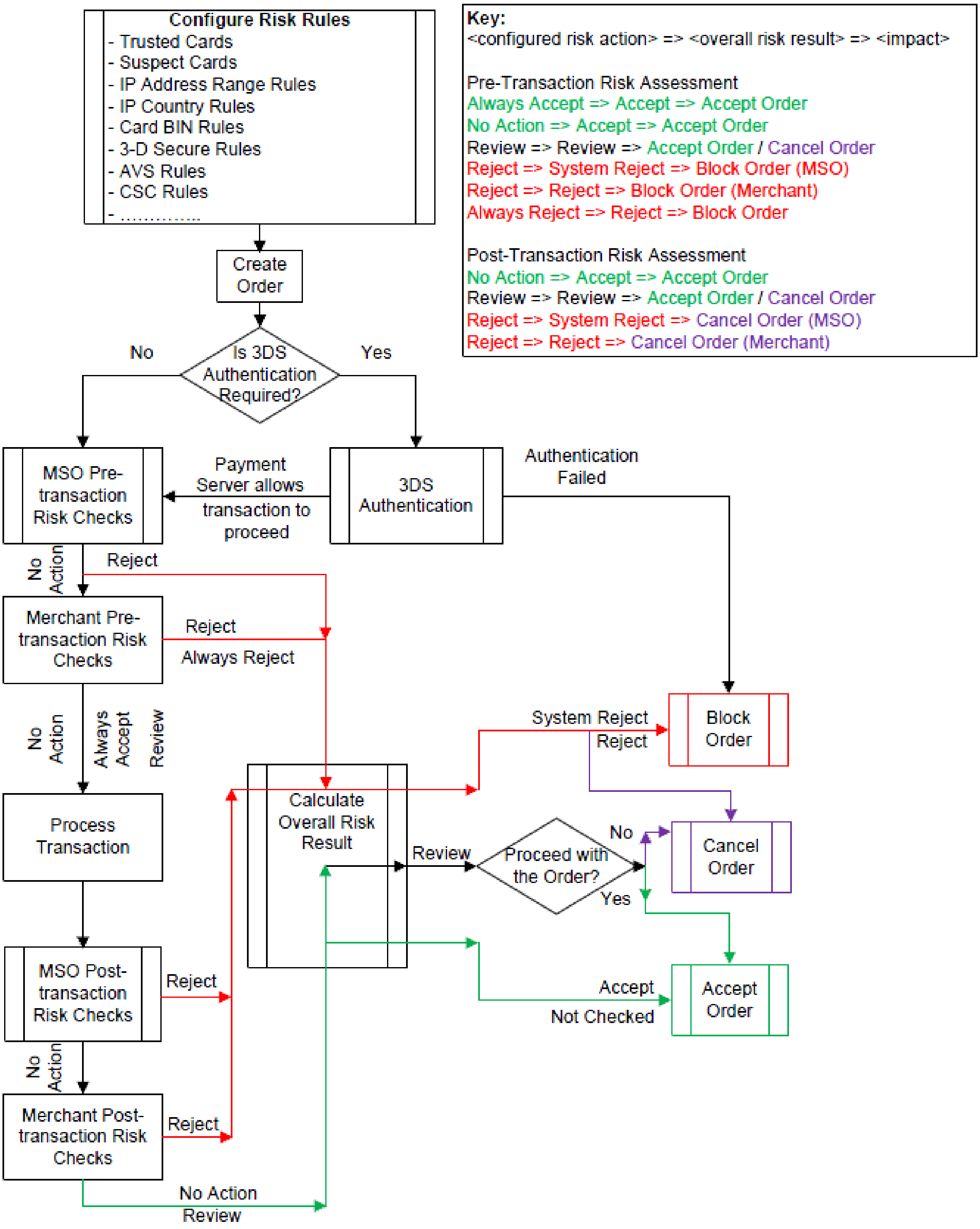
Note: Risk Management is applicable only to:

- 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 and merchant respectively.


Configure Risk Rules
- IP Address Range Rules
- IP Country Rules
- Card BIN Rules
- 3-D Secure Rules
- CSC Rules
- ....
Key:
<configured risk action> => <overall risk result> => <impact>
Pre-Transaction Risk Assessment
No Action => Accept => Accept Order
Reject => System Reject => Block Order
Post-Transaction Risk Assessment
No Action => Accept => Accept Order
Reject => System Reject => Cancel Order
Order Created
Is 3DS Authentication Required?
No
Yes
MSO Pre-transaction Risk Checks
Payment Server allows transaction to proceed
3DS Authentication
Authentication Failed
No Action
Reject
Process Transaction
Block Order
MSO Post-transaction Risk Checks
Reject
Calculate Overall Risk Result
System Reject
Cancel Order
No Action
Accept
Accept Order



Key:
<configured risk action> => <overall risk result> => <impact>

Pre-Transaction Risk Assessment
Always Accept => Accept => Accept Order
No Action => Accept => Accept Order
Review => Review => Accept Order / Cancel Order
Reject => System Reject => Block Order (MSO)
Reject => Reject => Block Order (Merchant)
Always Reject => Reject => Block Order

Post-Transaction Risk Assessment
No Action => Accept => Accept Order
Review => Review => Accept Order / Cancel Order
Reject => System Reject => Cancel Order (MSO)
Reject => Reject => Cancel Order (Merchant)

Dynamic Currency Conversion (DCC)

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

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.

Custom Payment Plans

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

Note: Applicable only to transactions using Mexican Peso currency.

A custom payment plan typically includes:

- Plan Name

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

  A auto-generated unique identifier for the payment plan. The Plan ID is unique across all Payment Plan Types for the merchant
- Installment Months (if applicable to the plan type)

  The installment terms in months configured for the payment plan.
- Deferral Months (if applicable to the plan type)

  The deferral terms in months configured for the payment plan.

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

Plan N

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

Generally there is a maximum of 24 installments.

Plan Amex

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

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

Plan Amex Inquiry

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

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

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.

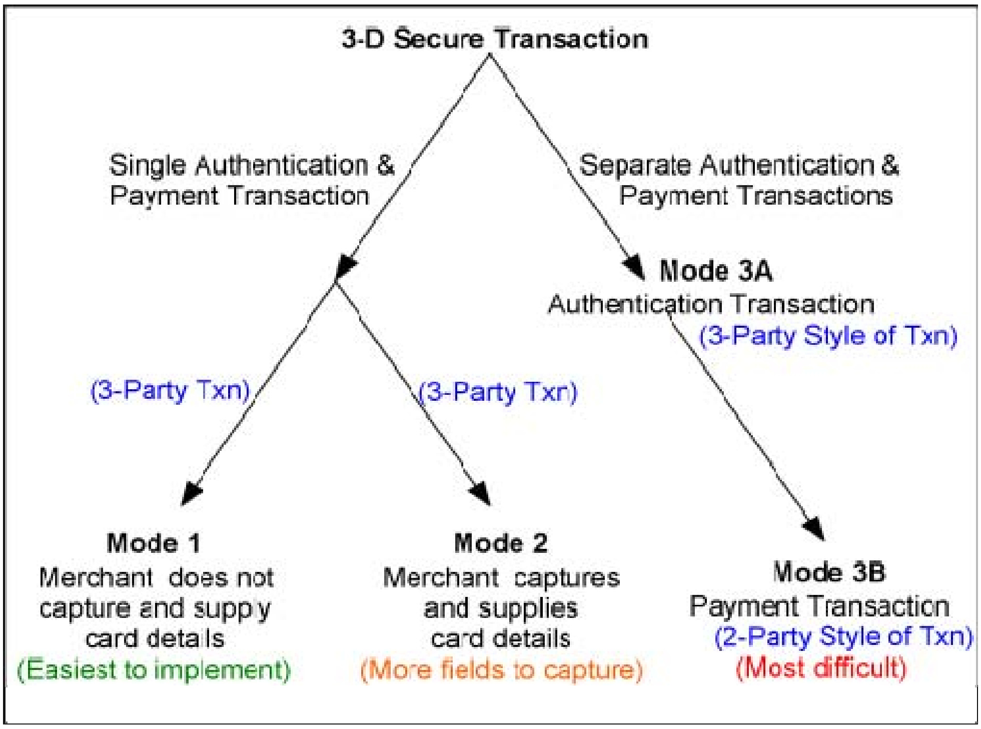
During a transaction, 3DS authentication allows the merchant to authenticate the cardholder by redirecting them to their card issuer where they enter a previously registered password.

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.

Payment Authentication 3-D Secure transaction modes

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


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

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

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

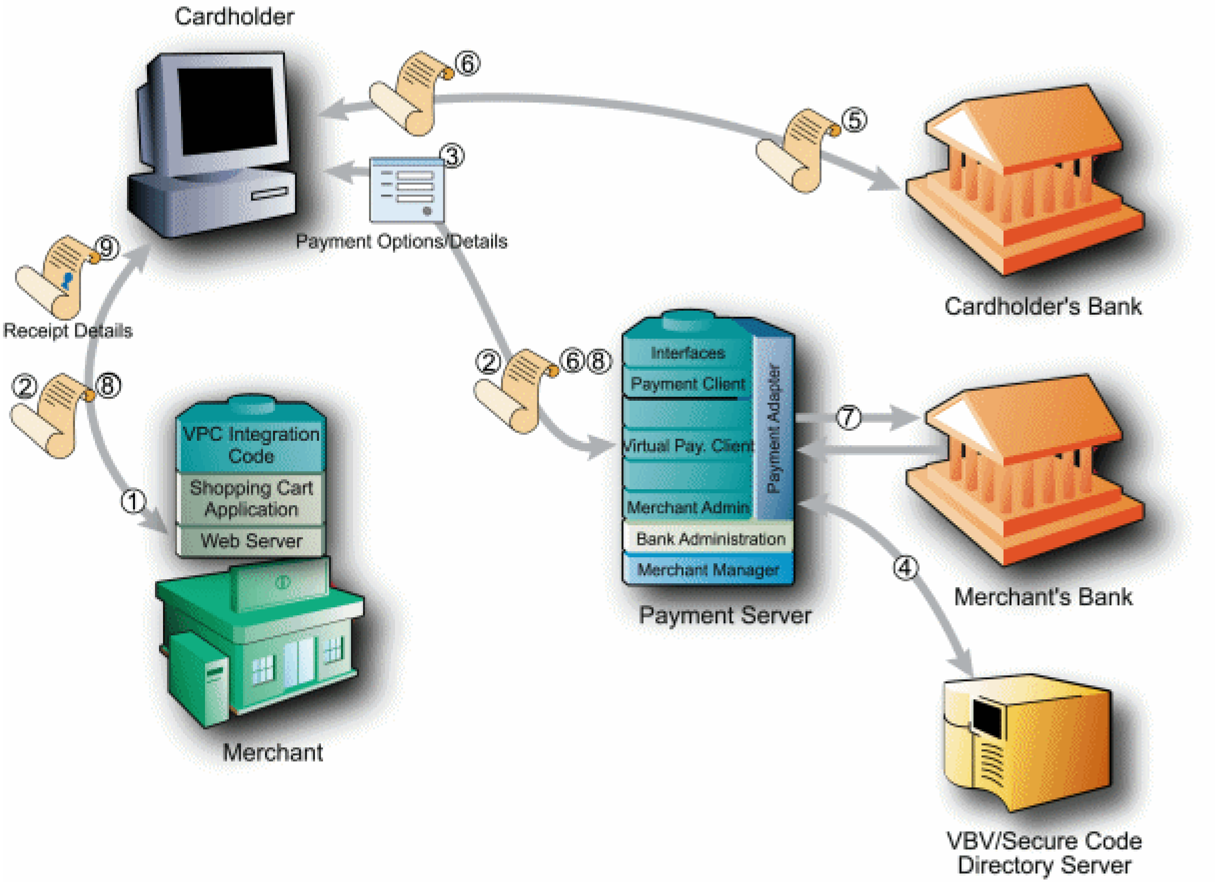
2 **Mode 2 - Combined 3-Party Authentication and Payment transaction, (merchant collects card details)** the merchant uses the Payment Server to perform the authentication and payment in the one transaction.

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

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

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

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



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

1. A cardholder browses the application, selects a product and enters their shipping details into the merchant's application at the checkout page.
2. The cardholder clicks a pay button and your application sends the payment Transaction Request to the Payment Server by redirecting the cardholder's Internet browser to the Payment Server.
3. The Payment Server prompts the cardholder for the card details.
4. If the card is a Visa or MasterCard, for example, the Payment Server then checks with the VBV or SecureCode Directory Server to determine if the card is enrolled in either the Verified by Visa™ (Visa 3-Domain Secure) or MasterCard SecureCode™ (MasterCard 3-Domain Secure) scheme.
   If the card is not enrolled in payment authentication scheme then go to Step 7.
   If the cardholder's card is registered in the payment authentication scheme, the Payment Server redirects the cardholder's browser to the card issuing site for authentication.
5. If the cardholder's card is registered in the payment authentication scheme, the Payment Server redirects the cardholder's browser to the card issuer's site for authentication. The card issuer's server displays the cardholder's secret message and the cardholder enters their secret password, which is checked against the Issuing bank's database.
6. At the completion of the authentication stage, the cardholder is redirected back to the Payment Server indicating whether or not the cardholder's password matched the password in the database.

   If the cardholder was not authenticated correctly, then the payment does not take place and the cardholder is redirected back to the merchant's site with a Transaction Response containing details to indicate the authentication failed - see step 8.
7. If the cardholder was authenticated correctly, or Payment Authentication did not occur the Payment Server continues with processing the transaction with the results of the authentication attempt.
8. The Payment Server then redirects the cardholder back to merchant's site with the Transaction Response. The Transaction Response contains the result of the transaction.
9. The application processes the Transaction Response and displays the receipt.

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

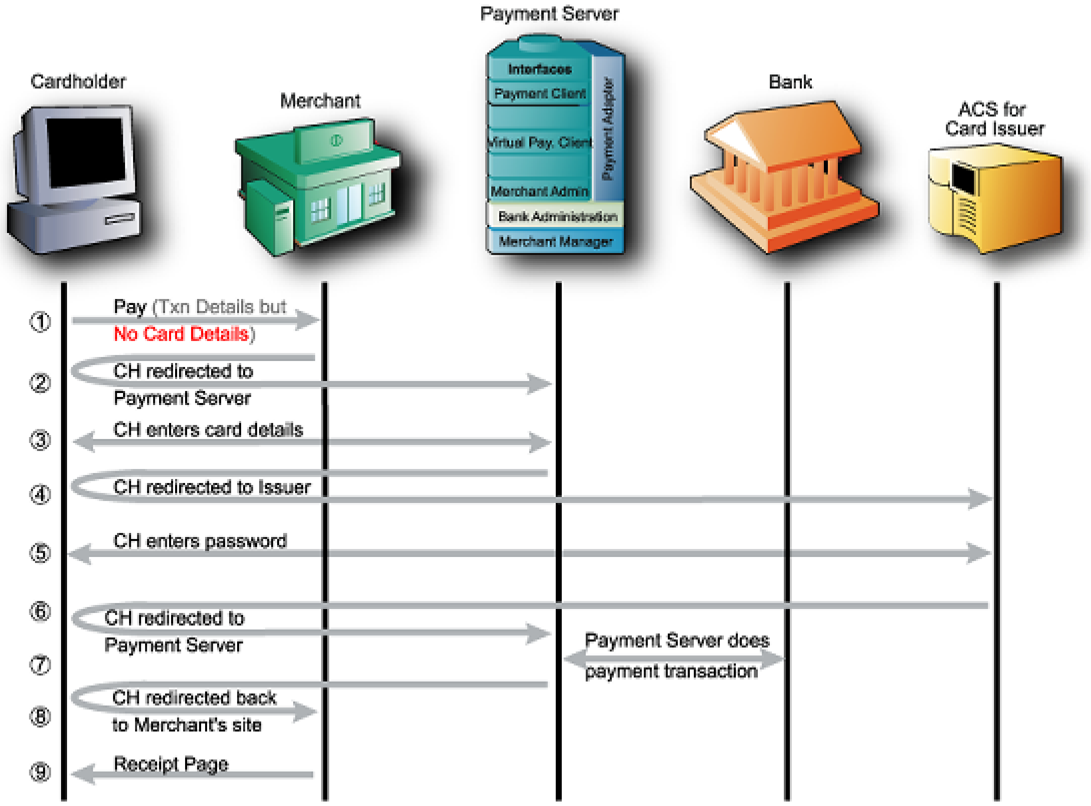
Mode 2 and Mode 3a are slight variations on the above information flow. In mode 2 and mode 3a the merchant collects the card details and passes them through, which means step 3 is eliminated.

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

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

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

Mode 1 - 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.

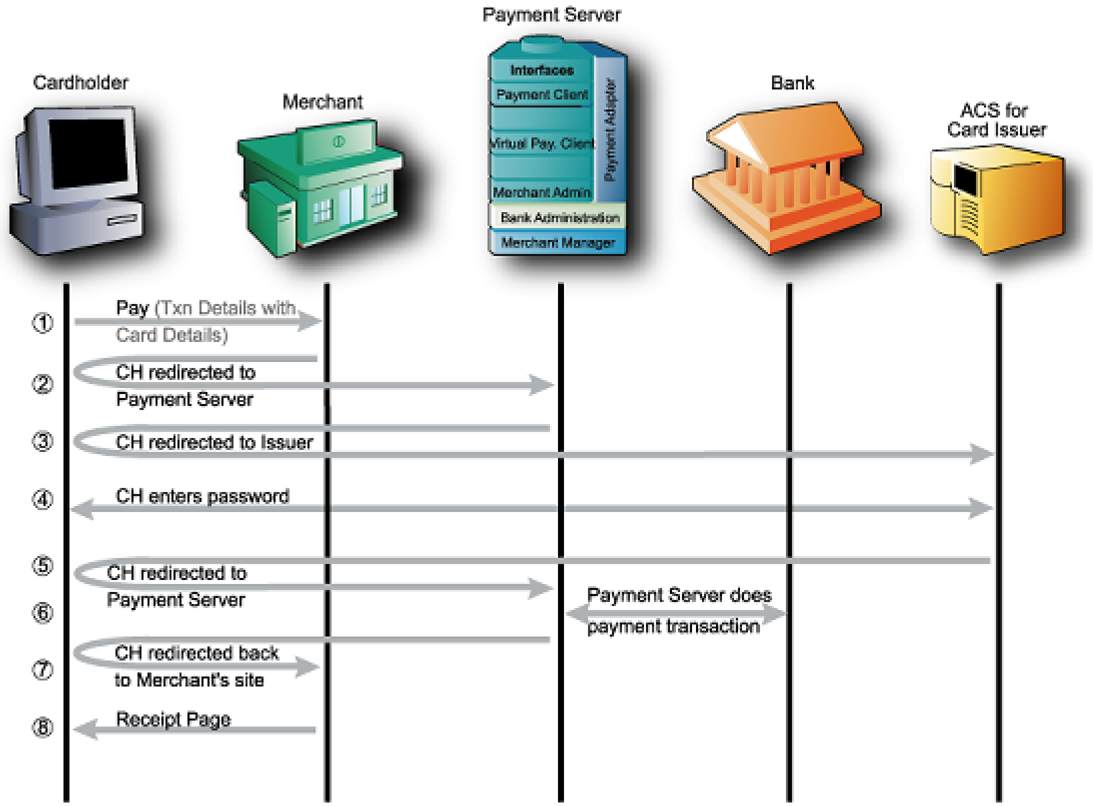
6 The cardholder is redirected back to the Payment Server and the card issuer returns an authentication message showing whether or not the cardholder's password matched the password in the card issuer system. If the Authentication failed, step 7 is bypassed and the cardholder is redirected back to the 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 and:

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


Payment Server
Interfaces
Payment Client
Virtual Pay. Client
Merchant Admin
Bank Administration
Merchant Manager
Payment Adaptor
Cardholder
Merchant
Bank
ACS for
Card Issuer
① Pay (Txn Details with Card Details)
② CH redirected to Payment Server
③ CH redirected to Issuer
④ CH enters password
⑤ CH redirected to Payment Server
⑥ Payment Server does payment transaction
⑦ CH redirected back to Merchant's site
⑧ Receipt Page

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 3-D Secure 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.

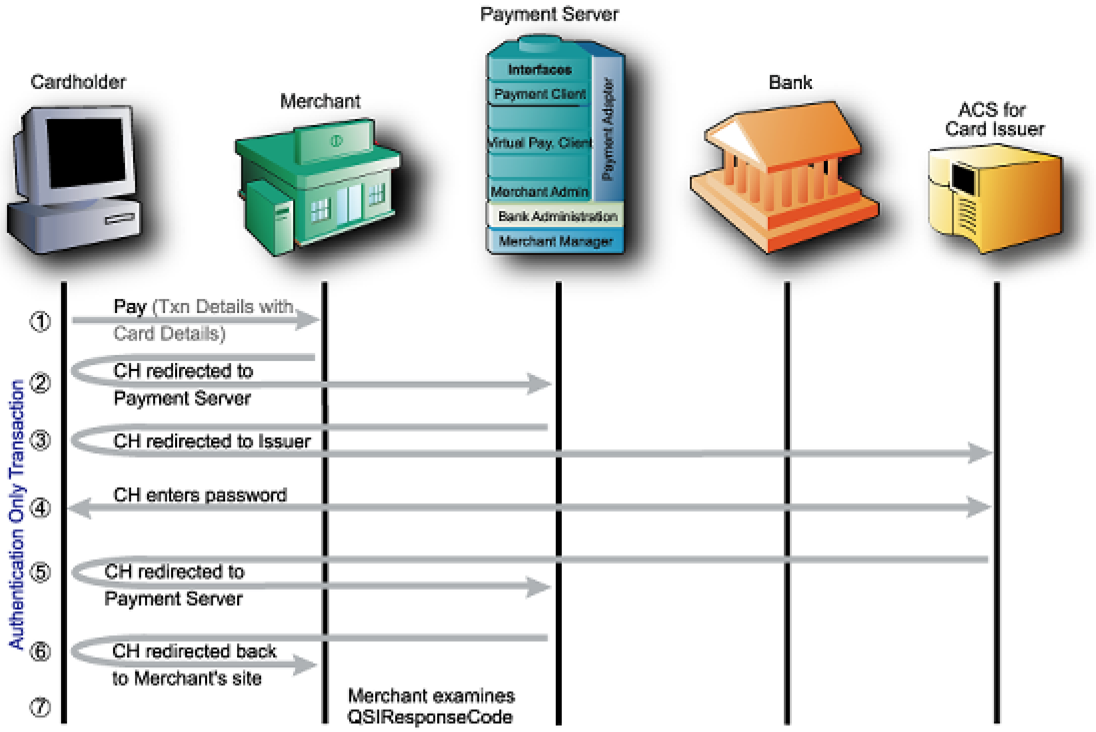
6 If the cardholder is authenticated correctly, the Payment Server continues with processing the payment part of the transaction.

7 The cardholder is redirected back to the merchant, where the receipt is passed back to the merchant and:

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.


Payment Server
Interfaces
Payment Client
Virtual Pay. Client
Merchant Admin
Payment Adaptor
Bank Administration
Merchant Manager
Cardholder
Merchant
Bank
ACS for Card Issuer
Authentication Only Transaction
① Pay (Txn Details with Card Details)
② CH redirected to Payment Server
③ CH redirected to Issuer
④ CH enters password
⑤ CH redirected to Payment Server
⑥ CH redirected back to Merchant's site
⑦ Merchant examines QSIResponseCode

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 3-D Secure 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.

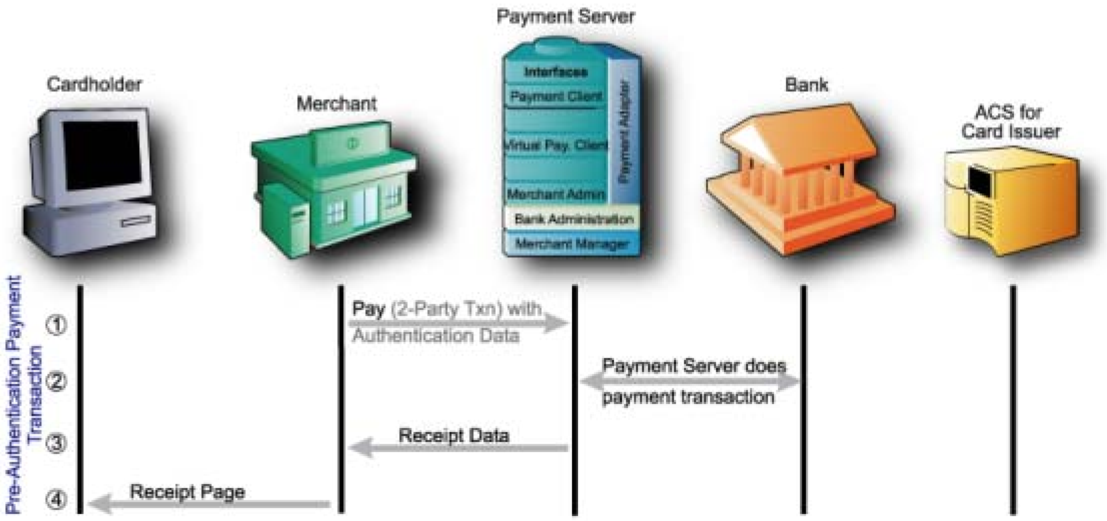
5 The cardholder is redirected back to the Payment Server and the card issuer sends an authentication message showing whether or not the cardholder's password matched the password in the Card Issuer system

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*** on page 37.



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.

CHAPTER 10

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 **t**his 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 Authorisation

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

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 transaction receipt. The search is performed on the key - *vpc_MerchTxnRef*, so the *vpc_MerchTxnRef* field must be a unique value.

Digital Receipts are stored by the Payment Server for up to 3 days only.

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 (3-D Secure) Authentication Only transactions.

AVS Only

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

With AVS Only, the Payment Server strips the value in the Authorisation transaction and substitutes a nominal transaction value (usually $0). The acquiring bank checks the card details with the issuing card institution to ensure they are correct. No funds at all are reserved on the card.

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

AMA Prior Authorised Transaction

The Prior Authorised Transaction command allows a merchant to resubmit a referred initial transaction with an authorisation code obtained from the Issuer. This transaction is not possible for Debit and EBT transactions.

The Prior Authorised Transaction must be run before the acquiring institution processes the batch containing the original transaction.

This transaction is only supported on the AMERICAN EXPRESS LAC and the MIGS S2I acquirers.

There are two ways to resubmit referred transactions.

1. Manually using Merchant Administration. This is the simplest method the merchant does not many referred transactions. For more information refer to the Merchant Administration User Guide.
2. Using the Prior Authorised Transaction command via the Virtual Payment Client to directly perform Referred Transactions from the merchant application.

CHAPTER 11

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 3-D Secure) 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?

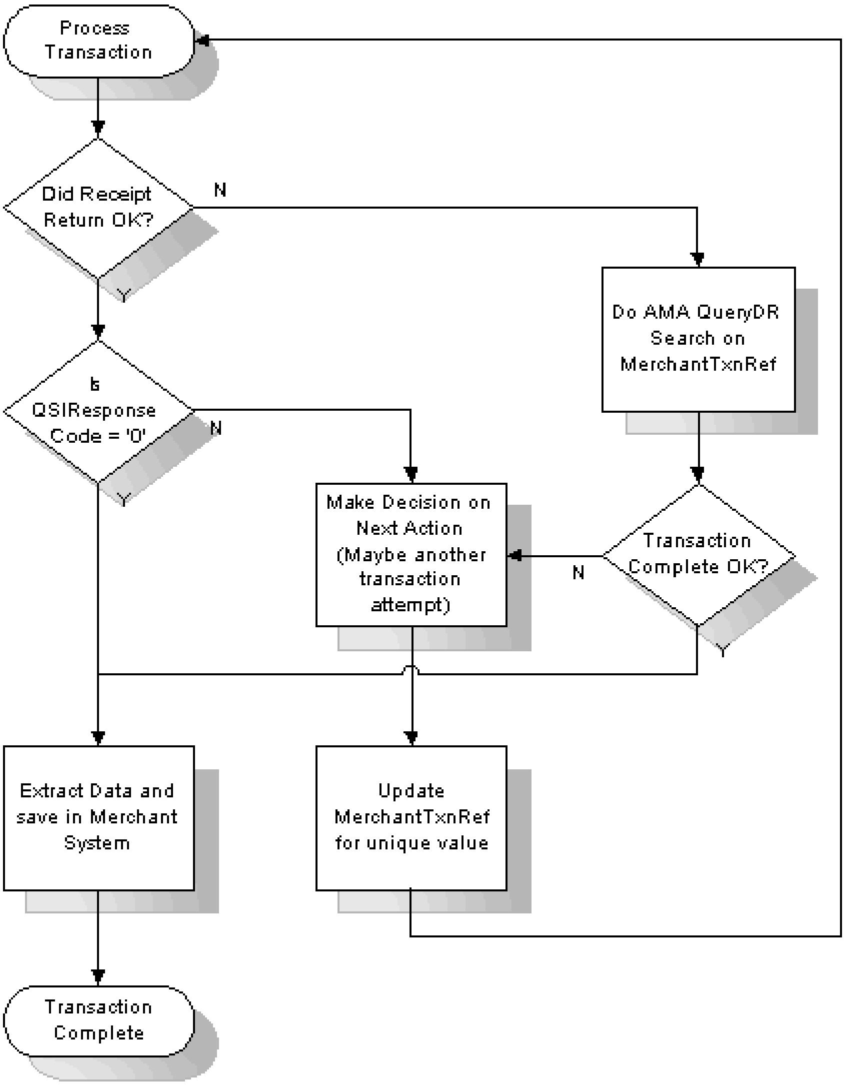
The Virtual Payment Client interface requires a cardholder's browser to support cookies for all 3-Party.

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.


Process Transaction
Did Receipt Return OK?
N
Y
Do AMA QueryDR Search on MerchantTxnRef
Is QSIResponse Code = '0'
N
Y
Make Decision on Next Action (Maybe another transaction attempt)
Transaction Complete OK?
N
Y
Extract Data and save in Merchant System
Update MerchantTxnRef for unique value
Transaction Complete

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 3-D Secure 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 Providers URL (or web site address) and a merchant profile. The merchant profile is a record of your details and privileges, which are 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, AuthorizeId 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.
- *AuthorizeId* 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 Transaction Request 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	▪ Expired card ▪ Account upgrade ▪ Portfolio sale ▪ Conversion	Obtain new account information before next billing cycle.
51	02	Try again later	▪ Over credit limit ▪ Insufficient funds	Recycle transaction 72 hours later.
05 14 51 54	03	Do not try again	▪ Account closed ▪ Fraudulent	Obtain another type of payment from customer.

CHAPTER 12

Appendix

PayPal Acquirer

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

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

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

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

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

Supported Transaction Types

PayPal supports the following transaction types:

Transaction Type	Supported Application	Notes
Authorisation	▪ 3-Party transaction (PayPal pages only)	An initial transaction that includes the merchant's website, Payment Server, and PayPal pages.
Capture	▪ Merchant Administration ▪ 2-Party AMA	A subsequent transaction on an authorisation transaction. The total amount captured (includes existing captures plus the amount to be captured) cannot exceed the order amount.
Partial Capture	▪ Merchant Administration ▪ 2-Party AMA	A subsequent transaction on an authorisation transaction.
Purchase	▪ 3-Party transaction (PayPal pages only)	An initial transaction that includes the merchant's website, Payment Server, and PayPal pages.
Refund	▪ Merchant Administration ▪ 2-Party AMA	A subsequent transaction on a capture or purchase transaction. The following rules/limitations apply: ▪ If a single capture exists on an order, or the merchant is a purchase mode merchant, the merchant, via 2-Party AMA, does not need to specify the financial transaction to refund. ▪ If a single capture exists on an order, or the merchant is a purchase mode merchant, the merchant may process the refund via Merchant Administration. ▪ If more than one capture exists on an order, the merchant, via 2-Party AMA, must include the specific capture financial transaction ID to refund. ▪ If the merchant is refunding a specific capture transaction, the amount of the refund cannot exceed the amount of the specific capture transaction.
Partial Refund	▪ Merchant Administration ▪ 2-Party AMA	A subsequent transaction on a capture or purchase transaction. **Note**: The rules outlined for Refund apply to Partial Refunds also.
Void Authorisation	▪ 2-Party AMA	A subsequent transaction on an authorisation transaction.
QueryDR	▪ 2-Party AMA	A transaction to get the current status of the transaction.
Standalone Capture	▪ Not Supported	-
Standalone Refund	▪ Not Supported	-

Credit Payment	▪ Not Supported	-
Payment Plan	▪ Not Supported	-

Common Fields Between PayPal and the Payment Server

PayPal Field	Payment Server Field	Notes
Invoice ID	Order Reference [Shopping Transaction Number]	Invoice ID is a PayPal identifier which uniquely identifies an order. It comprises of the Payment Server Order Reference and the Payment Server Shopping Transaction Number. PayPal Invoice ID = Payment Server Order Reference [Payment Server Shopping Transaction Number] Since it's not mandatory for the Payment Server Order Reference to be unique, the Shopping Transaction Number is suffixed to achieve uniqueness.
PayPal Payer ID	Card Number	Payer ID is the email address of the purchaser set up with PayPal and is a unique identifier for the purchaser. This is the equivalent to the Payment Server cardholder's card number for Credit payment methods. The Payer ID can be accessed via card number fields (e.g. on the Order Search or in the Order Download). The Payer ID is also displayed in the *Order Details* page in Merchant Administration. PayPal Payer ID = Payment Server Card Number
PayPal Transaction ID	RRN	PayPal Transaction ID is a PayPal-supplied identifier for the transaction. A transaction can include authorisation, capture, refund, etc. This may be used by the customer to search for the transaction at PayPal. The PayPal Transaction ID is the equivalent of vpc_ReceiptNo, RRN, defined in Payment Server. The merchant can view the PayPal Transaction ID in the *Financial Transaction Details* page in Merchant Administration, PayPal's *Transaction Details* page, and QueryDR functionality. The purchaser can view the PayPal Transaction ID in PayPal's *Review Payment* page. PayPal Transaction ID = Payment Server RRN
PayPal.ACK PayPal.PaymentStatus PayPal.PendingReason	Acquirer Response Text	Acquirer Response Text is the response code returned by PayPal in the text form. It is used instead of the acquirer response code which is returned for other credit acquirers. The format is: **PayPal.ACK** +" status: " + **PayPal.PaymentStatus** + " reason: " + **PayPal.PendingReason** = Payment Server Acquirer Response Text For example, Success : Pending: Authorization

Unsupported Features

The following features are currently **NOT** supported for PayPal:

- 3DS Authentication
- Address Verification Service
- Card Security Code
- Excessive Refund
- Settlement, Reconciliation, Batch and Batch Closure
- Card details in Transaction Request

How to Set up a PayPal Account with TNS

To enable PayPal as an additional payment option to your customers, you must first set up a merchant account with PayPal.

Please follow the instructions detailed on this page to enable PayPal through your Payment Provider.

Step 1: Set Up a Verified PayPal Business Account

Customers who don't have an existing PayPal account:

1 Go to ***https://www.paypal.com*** https://www.paypal.com

2 Click **Sign Up**.

3 Set up an account for Business Owners.

4 Follow the instructions on the PayPal site.

Customers who already have a Personal or Premier account:

1 Go to ***https://www.paypal.com*** https://www.paypal.com

2 Click the **Upgrade your Account** link.

3 Click the **Upgrade Now** button.

4 Choose to upgrade to a Business account and follow instructions to complete the upgrade.

Become a Verified member:

1 If you haven't already, add a bank account to become a Verified member.

2 PayPal will deposit two small amounts into your account. This process can take 3 to 5 days.

3 Once you see the two deposits in your bank account, log into your PayPal account and click **Complete Bank Setup** on the Account Overview page (found in the **To do list** section.).

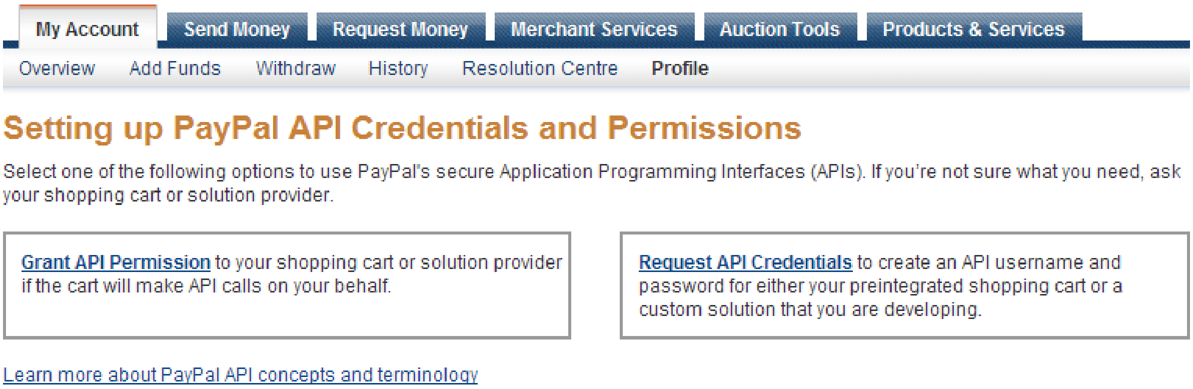
4 Enter the two deposit amounts as prompted. The process is then complete and your PayPal account is verified.

Step 2: Third-party authentication: Grant your cart the appropriate API authentication permissions

1 Log in to your PayPal account and click the **Profile** subtab.

2 Click the **API Access** link in the Account Information column.

3 Click the **Grant API Permission** link.



Setting up PayPal API Credentials and Permissions

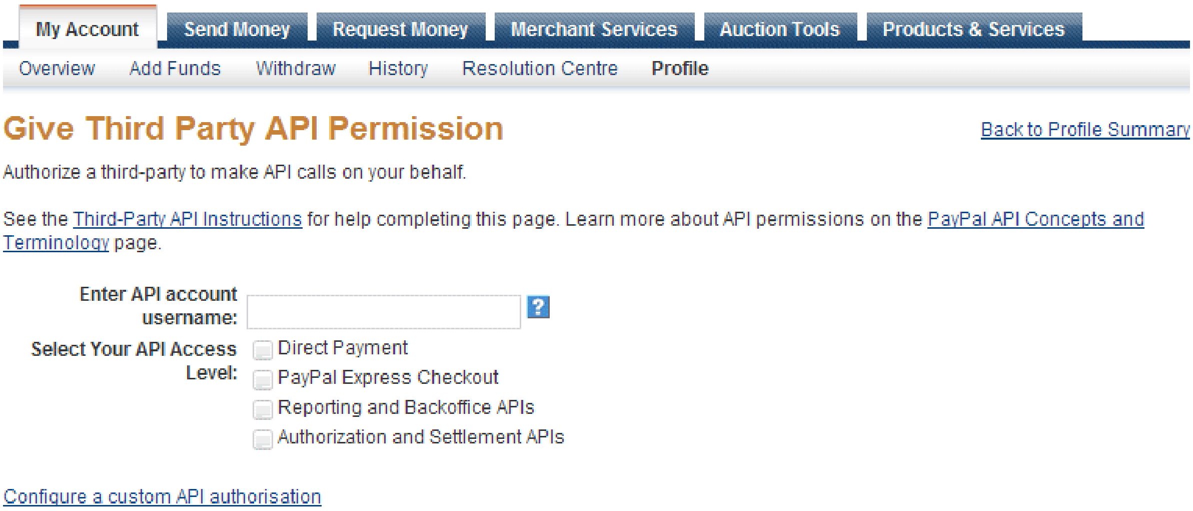
Select one of the following options to use PayPal's secure Application Programming Interfaces (APIs). If you're not sure what you need, ask your shopping cart or solution provider.

Grant API Permission to your shopping cart or solution provider if the cart will make API calls on your behalf.

Request API Credentials to create an API username and password for either your preintegrated shopping cart or a custom solution that you are developing.

Learn more about PayPal API concepts and terminology

4 In the **Enter API account username** field, enter the API Account username provided by TNS. This username will be provided in the merchant setup email.



Give Third Party API Permission

Back to Profile Summary

Authorize a third-party to make API calls on your behalf.

See the Third-Party API Instructions for help completing this page. Learn more about API permissions on the PayPal API Concepts and Terminology page.

Enter API account username:

Select Your API Access Level:
□ Direct Payment
□ PayPal Express Checkout
□ Reporting and Backoffice APIs
□ Authorization and Settlement APIs

Configure a custom API authorisation

5 Check **PayPal Express Checkout** and any other APIs the customer needs to access:

- **Reporting and Backoffice APIs**
- **Authorization and Settlement APIs**

6 Click the **Submit** button then **Give Permission** button on the following page to finalise the process.

How to Test a PayPal Transaction

Performing test transactions allows you to test your integration, so that you won't encounter problems when processing real transactions. The following steps take you through the process of testing PayPal transaction; however, if you still encounter issues do not hesitate to contact TNS Customer Support.

1. Create a Sandbox account with PayPal at ***https://developer.paypal.com*** https://developer.paypal.com. For information, see PayPal Sandbox User Guide.
2. Create a Sandbox merchant account, say merchant_name@your_merchant_website.com. Note that this need not be a real email address.
3. Create a Sandbox payer account, say payer_1@your_merchant_website.com. Note that this need not be a real email address.
4. Log into your TEST merchant profile set up with TNS to perform a test transaction. The TEST merchant profile is associated with the email address of your Sandbox merchant account, that is merchant_name@your_merchant_website.com
5. Log into PayPal Sandbox to grant TNS third party api permissions on your PayPal Sandbox merchant account. See Step 2 in ***How to Set up a PayPal Account with TNS*** on page 97.

   Note: The **API Account Username** used for the PayPal Sandbox merchant account differs from the PayPal production merchant account.

6. Log in to ***https://developer.paypal.com*** https://developer.paypal.com. PayPal requires you to be logged into your PayPal Sandbox merchant account during testing. From the Sandbox, you can also see the transactions as received by PayPal.
7. Update your application integration to use PayPal fields. For more information, see VPC Integration Guide and VPC Reference Guide.
8. Use the TEST account on your merchant profile to perform an initial transaction.
9. To test subsequent transactions such as Capture, Refunds, etc., you can use Merchant Administration or the VPC 2-Party gateway.

Index