DataCash MiGS Virtual Payment Client Integration Guide

August 2014

Software version: MR 29.6

Contents

1 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 that you use (either 3-Party or 2-Party).

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 cardholders' card details on your behalf. The Payment Server's payment pages could be Bank branded 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 website and pass them through with the other transactional details. If this is done, the Payment Server does not display any third 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 initiative. To do this, you 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 initiatives of MasterCard® SecureCode™, Verified by Visa™ and JCB J/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 completes the transaction.

The three parties involved in a 3-Party transaction are the merchant, the Payment Provider and the cardholder. The cardholder's browser provides the Redirect method to communicate the information between the merchant and the Payment Provider. This is an asynchronous connection and the cardholder leaves your website to go to the Payment Server, which means the transaction is broken or disrupted into two distinct sessions, namely 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 model requires you to securely collect the cardholders' 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 numbers and details.

The 2-Party model does not allow you to implement the 3-D Secure initiatives of MasterCard® SecureCode™, Verified by Visa™ and JCB J/Secure™.

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

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

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

Selection Guidelines for Integration Models

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

When to use 3-Party Payments

Consider using 3-Party Payments if:

- You are integrating a web browser-based application only. Call centres, IVRs and other applications cannot use this transaction mode.

- You want to, either now or in the future, increase security by using 3-D Secure authentication (for example, MasterCard® SecureCode™, Verified by Visa™ and JCB J/Secure™).

- It is acceptable to have the cardholders' browsers redirected away from your website to the Payment Server.

- You want the Payment Provider to collect and manage the cardholders' 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 in 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 (e.g. VISA AIS, MasterCard SDP).

- 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, MasterCard® SecureCode™, Verified by Visa™ and JCB J/Secure™).

  For more information see *Payment Authentication* on page 43).

- You do not want the cardholders' browsers to be redirected away from your website 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 models if any of the following are true:

- You want to use a combination of 3-Party for internet and 2-Party for call centre/IVR/other applications.

- You have a web application in which you want to perform some form of repeat payments, 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 instalment 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 model transactions.

> **Note:** 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, Visa or JCB. Other transactions that don't use 3-D Secure authentication can be performed using 2-Party transactions.

> **Note:** Advanced Merchant Administration functions such as captures, refunds, voids and queries all use the 2-Party model 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.

About Merchant IDs

To use Merchant Administration, a merchant profile is required. The profile is a record of merchant details and the permitted functionality that the merchant has within the MA portal; all details are stored on the MiGS Server. Two types of merchant profile are created through the bank's enrolment process:

- **TEST merchant profile**—allows merchants, within the test facility, to perform transactions against an emulator of the bank's transaction processing system. This profile will always exist for testing purposes. To access this facility, precede the merchant ID with the word TEST (i.e. MERCHANT01 becomes TESTMERCHANT01).

  In TEST mode only, the decimal value will determine the Acquirer/Issuer Response Code (e.g. 5995 will return an Acquirer Response Code of 95). Once you move to production the response will be provided by the Issuer and has nothing to do with the Amount's decimal value. Use an Amount that is a multiple of 100 to simulate an approved transaction during TEST mode (e.g. 100, 43500, 700).

  There are also test cards that will use the expiry date to determine the Response Code instead of the submitted Amount. Your MiGS contact can provide the set of test card numbers.

- **PRODUCTION merchant profile**—activates merchants within the production system, allowing them to process transactions directly against the MiGS live transaction processing system. This profile is only activated once testing has been deemed sufficient by the bank.

Contact your Payment Service Provider when you have completed your testing. The Production merchant ID must be enabled by the PSP first.

> **Note:** The Access Code / Secure Hash Secret for the Test Merchant ID and the Production Merchant ID are different.

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 Integration Reference
- Example code for your site (written in ASP, JSP, PHP and Perl)
- Test Card setup document

> **Note:** Only Virtual Payment Client Integrations are now supported for new integrations.

Determine Your Integration Model

You must choose either:
- 3-Party Payment Integration Model, or
- 2-Party Payment Integration Model, or
- Combination of 2-Party and 3-Party Integration.

For more information, see *Integration Models and Communication Methods* on page 4.

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.

Determine Any Advanced Functionality

The available advanced functionality includes:

- 3-D Secure authentication. See *Securing Your Payments* on page 14).
- Capture
- Refund
- Voids
- QueryDR.

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 the Merchant ID on the Payment Server.

- **Secure Hash Secret only**

  The Secure Hash Secret is a key used as the initial piece of encryption data to create a Secure Hash. This ensures transaction data is not tampered with while in transit to the Virtual Payment Client.

Your access code and secure hash secret can be found in Merchant Administration on the Configuration Details page of the Admin menu option.

To retrieve your Access Code and Secure Hash Secret:

1 Ask your Payment Service Provider for your MSO ID and log in details.
   Login to the new Merchant Administration website as Administrator:
   https://migs.mastercard.com.au/ma/<MSO ID>

2 Select **Admin > Operators**.

3 Click on the *Create a new Merchant Administration Operator* link.
   Ensure that you enable **Modify the merchant configuration**.

4 Click **Submit**.

5 Logout, then login again as the new Operator.

6 Select **Admin > Configuration Details**.

You will see the **Access Code** and **Secure Hash Secret**.

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 payment 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, *Handle Session Variables* on page 29).

  Session variables are not required when using the Direct or 2-Party communication method as the session is maintained 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 (vpc_MerchTxnRef)* on page 11.

  The Merchant Transaction Reference is primarily used in the QueryDR function, which is used for reconciling timed out transactions.

MiGS Payment Server IP Addresses

You must ensure your system and firewall are set up to access both the primary and fallback (DR) MiGS Payment Server IP addresses. This is to ensure that when MiGS switches to the fallback address, your firewall will continue to permit connectivity to the MiGS Payment Server.

It is highly recommended that you validate the SSL certificate of the Payment Server—for both the primary and fallback (DR) servers—whenever you connect to the Payment Server

The MiGS Payment Server IP addresses are:

- MiGS primary IP address: 203.42.65.51
- MiGS fallback (DR) IP address: 216.119.208.69

Design and Implement the Integration

You are now ready to 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 Integration Reference 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 a list of card numbers to use for testing, refer to Appendix A.

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 the end-to-end functionality works correctly, including successful settlement of funds from your financial institution.

Remember you can always refund or void these test transactions.

Go Live

Once you are satisfied that your integration works correctly, 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.

Commence Live Online Payments

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

Virtual Payment Client Integration Guidelines

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

Reference Fields

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

Merchant Transaction Reference (vpc_MerchTxnRef)

The **vpc_MerchTxnRef** field is a unique identifier that the merchant assigns to each transaction. This unique value is used by the merchant to query the Payment Server database to retrieve a copy of a lost/missing transaction receipt using a 2-Party QueryDR function. This 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.

Order Information/Order Reference

vpc_OrderInfo is an identifier provided by you to identify the order on the Payment Server database. This value is displayed in the Merchant Administration portal when manually searching orders. It can be an order number, an invoice number, or a shopping cart number.

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

Ensuring Successful Payments

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

An issue that merchants have to deal with when implementing payments solutions is 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 are paid you can:

- Where possible, implement the 3-Domain Secure authentication services of the card schemes. See *Securing Your Payments* on page 14).

- 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 cardholders' orders. Also as volumes grow, the risk may become significant as would the time and cost involved completing the task.

Ensure Correct Character Encoding

The 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**.

Note: It is critical that you verify that the Secure Hash of the Digital Receipt / Transaction Response is correct.

See *Detect alteration of Requests and Responses using Secure Hash* on page 18.

2   Securing Your Payments

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

Protecting Cardholder Information Using SSL

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

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

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

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

> **Note:** SSL should also be used for 3-Party. This avoids the possible browser alert message indicating that the cardholder is being redirected to an unsecured site. This can happen when the cardholder's browser is being redirected back to the merchant's website 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 padlock symbol signifies that communication with the Payment Server is encrypted and secure.

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

Using 3-D Secure Payment Authentications

3-D Secure Payment Authentication contains MasterCard® SecureCode™, Verified by Visa™, JCB J/Secure™, American Express SafeKey™ and Diners Club ProtectBuy^{SM} 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 authentication 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 and Payment transaction see *3-Party Payment Information Flow*.

If the merchant is using 3-D Secure payments, the payment page displays the appropriate 3-D Secure logo.

The following example shows a MasterCard SecureCode request screen:

The following example shows the Verified by Visa request screen:

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 or 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 the values input to the original Transaction Request.

Check for suspect transactions

Common things to look out for are:

- Use of free or anonymous email 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 website 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 that you 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—for both the primary and fallback (DR) servers—whenever you connect to the Payment Server. The Payment Server SSL certificate is issued by an industry standard Certificate Authority whose root certificate should already be available in your web environment.

Note: Consult a web developer if you are not familiar with validating SSL certificates or exporting certificates from websites. Always ensure the server is a trusted source.

Additional Features for 3-Party Transactions

The following features apply to 3-Party transactions only.

Detect alteration of Requests and Responses using Secure Hash

MiGS supports both MD5 and SHA-256 secure hash methods, however it is strongly recommended that new merchant integrations use SHA-256. MD5 is supported for existing merchant integrations.

The Virtual Payment Client uses a secure hash in transaction security as it is used to detect whether the Transaction Request and Transaction Response have 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 is 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 secure hash, which is appended to the Transaction Request. Because the Payment Server is the only other entity apart from your application that knows your Secure Hash Secret, it is the only other entity that can recreate the same secure hash:

- If the secure hash recreated by the Payment Server is equal to the secure hash sent in the Transaction Request, it means that the Transaction Request has not been tampered with. The Payment Server will continue to process the payment.

- If the secure hash recreated by the Payment Server does not equal the secure hash sent in the Transaction Request it can be assumed the data has changed in transit. The Payment Server will not process the payment and will 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 a 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.

For more information on Secure Hash Secret, see *Store Secure Hash Secret Securely* on page 18).

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 via the Configuration Details page of the Admin option. For more information, refer to your Merchant Administration User Guide.

3   Integrating 2-Party Payments

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

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

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

2-Party Payments Information Flow

The following is the information flow in a 2-Party transaction:
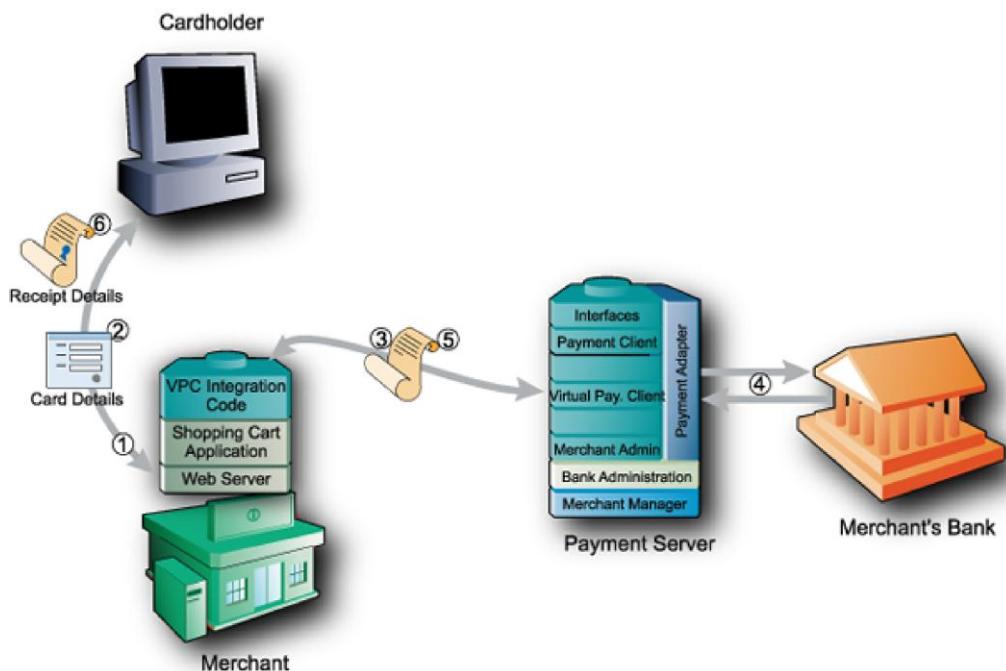
1. A cardholder decides to make a purchase and provides their card details directly to your online store.

2. Your application collects the details of the cardholder's order.

3. In addition it formulates the Transaction Request and sends it using a HTTPS POST over the Internet to the Payment Server via the Virtual Payment Client.

4. The Payment Server passes the transaction to the merchant's acquirer bank for processing.

5. After processing, the Payment Server generates a Transaction Response and passes it via the Virtual Payment Client to your online store. The Transaction Response shows whether the transaction was successful. The results can be stored by you for future reference.

6. A receipt is generated and either immediately passed to the cardholder or included when shipping the goods.

What the Cardholder Sees

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.

The cardholder is presented with two pages:

1. The merchant's application checkout page.

2. The merchant's application receipt page.

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

Merchant's Application Checkout page   Merchant's Application Receipt page

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.

4   Integrating 3-Party Payments

This section describes the information flow and integration model for 3-Party Payments. The three 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 completes the transaction.

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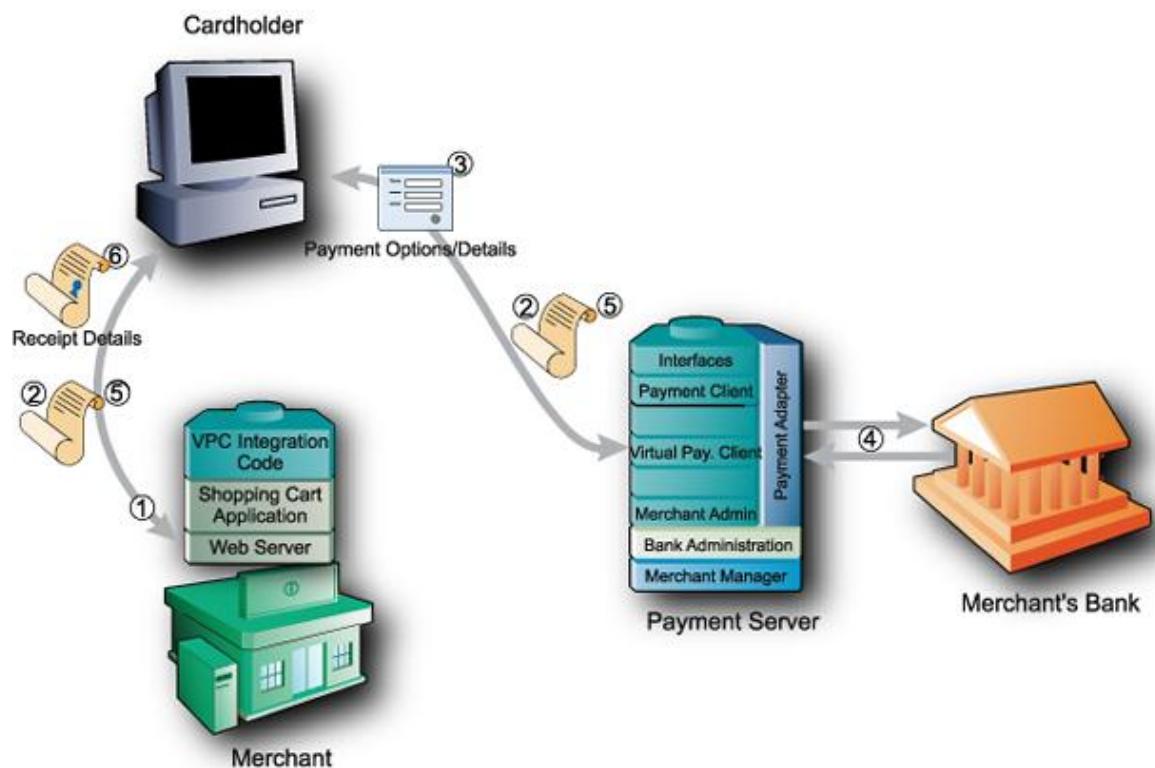
The following is the information flow in a 3-Party transaction:

1. A cardholder browses your online store, selects a product and enters their shipping details into your 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 date, 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, 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.

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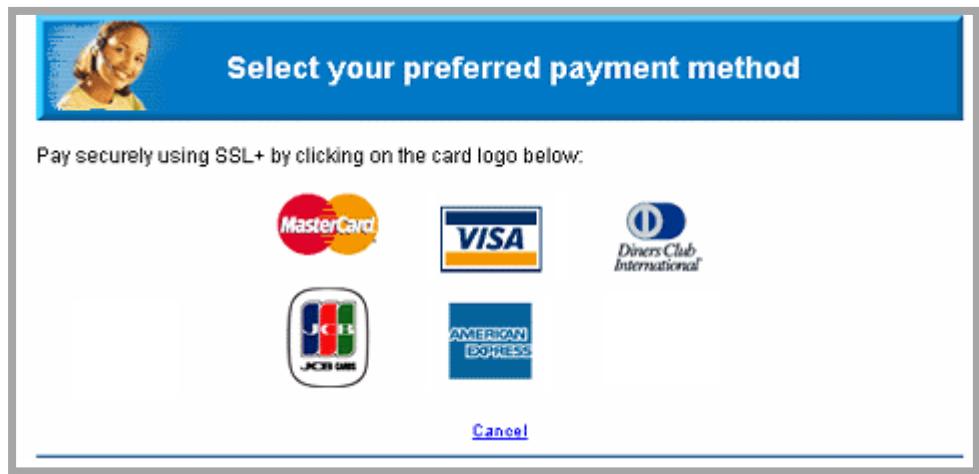
The payment options page presents the cardholder with the card types the merchant accepts. The cardholder clicks a card type and proceeds to the Payment Details web page.

3. The Payment Server's payment details page.

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

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.

Please wait while your payment is processed

Please wait...

The server is processing your payment using MasterCard for the value of AUD $123.00.
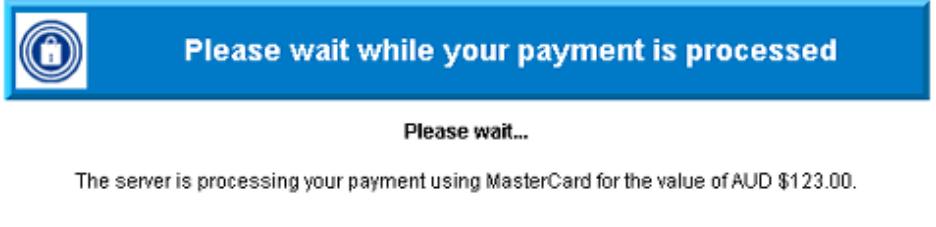
5. The Payment Server's redirection page.

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:

Transaction Results

Result of your transaction:

Your payment has been **approved**.

Please wait while you are redirected back to the merchant...

A declined transaction would appear as follows:

Transaction Results

Result of your transaction:

Your payment was NOT successful. The card you used has insufficient funds to cover the transaction.
Please wait while you are redirected back to the merchant...

6. The merchant's application receipt page.

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 secure hash signature
- Send it with the Transaction Request, and
- Check that the secure hash signature in the Transaction Response is valid for the received data.

To do this you need to do the following:

Handle a Transaction Request

1. Add any variables required by the application to re-create the session before the Transaction Request has been processed. These should be included in the secure hash signature.

2. Collect the minimum required information for a Transaction Request. This includes your MerchTxnRef, Merchant ID, an Order Information field, the Transaction Amount, the Locale and the Return URL to which the Payment Server needs to redirect the cardholder.

   You may require additional information fields when using optional features.

3. Formulate a Transaction Request using the fields outlined above.

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

An example of the start of a Transaction Request is:

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

What the Payment Server does

When a Transaction Request arrives at the Payment Server, the Payment Server checks the digital secure hash signature on the Digital Receipt, and:

- If the digital signature is correct, it decrypts the encrypted Digital Receipt data, and:
  - 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

- Sends back error messages, for example if there is a communication error in the banking network and the transaction cannot proceed.

- If the digital signature is incorrect:

  - Returns the cardholder to the merchant with a Transaction Response with an error message indicating the signature was invalid in the Transaction Request. No payment takes place.

Handle a Transaction Response

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

```
http://Merchant_Site_URL/Receipt.asp?vpc_AVSResultCode=Unsupported&vpc_Ac
qAVSRespCode=Unsupported&vpc_AcqCSCRespCode=Unsupported&vpc_AcqResponseCo
de=00&vpc_Amount=100&vpc_AuthorizeId=020072&vpc_BatchNo=20051209&vpc_CSCR
esultCode=Unsupported&vpc_Card=MC&vp
```

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

- Incorrect secure hash signatures

- Successful transactions

  - If **vpc_TxnResponseCode** code is equal to '**0**' then the transaction was completed successfully and you can display a receipt to the cardholder.

- Declined transactions

  - If **vpc_TxnResponseCode** is equal to '**1**', '**2**', '**3**', '**4**', or '**5**' the transaction has been declined and this needs to be conveyed back to the cardholder.

- Error Conditions –

  - If **vpc_TxnResponseCode** equals '**7**' or '**8**' an error has occurred

  - Other values may also 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 be returned 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.

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.

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 five session variables using any name that your application needs, providing:

- They conform to HTTP or 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_**.
  - With an MD5 secure hash, these variables must be present in the Transaction Request before the MD5 signature is calculated and appended to it.
  - With an SHA-256 HMAC secure hash, you must not use session variables. For more information, see **Creating a SHA-256 HMAC Secure Hash** in the Virtual Payment Client Integration 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.

Additional 3-Party Functionality

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

- Where the Payment Server collects the cardholder's payment details from a digital wallet, for example, MasterPass
- Where the merchant collects the cardholder's card type
- Where the merchant collects all the cardholder's card details
- Where the merchant is enabled for 3-D Secure. For more information, see *Using 3-D Secure Payment Authentications* on page 15.

3-Party Payments where the Payment Server collects the cardholder's payment details from a digital wallet

In 3-Party payments, you can allow the Payment Server to collect the card details of a customer from a digital wallet such as MasterPass Online. To achieve this, your Payment Provider must enable and configure you to offer the digital wallet as an option to your customers.

You can consider the following options when redirecting the customer to a digital wallet:

- The customer chooses to provide their card details using the digital wallet on your website.

  With this option, the customer is redirected to the digital wallet by bypassing all the Payment Server payment pages. This can be helpful if your website has already determined that the customer wishes to pay using the digital wallet. You can achieve this by setting the Card Source field to the digital wallet in the Transaction Request. For example, Card Source=MASTERPASS

  Note: The Card Source field takes precedence over the Card field when processing transactions. For example, if Card Source=SomeDigitalWallet and Card=SomeCard then the Card field is ignored and the customer is redirected to the digital wallet.

- The customer chooses to provide their card details by selecting the digital wallet on the Payment Server card selection page.

  With this option, the Card Source field must not be provided in the Transaction Request. The digital wallet is presented as an option on the Payment Server card selection page.

3-Party Payments using MasterPass Online

MasterPass Online is a digital wallet that allows customers to store details of one or more credit cards in a secure server. The customer can also choose to store other details such as billing address and shipping address. Merchants enabled for MasterPass Online can allow the Payment Server to redirect their customers to the MasterPass Online website to select their payment and shipping address details.

As described in the section above, you can choose to redirect the customer directly to the MasterPass Online website by setting the Card Source field to MASTERPASS. If you do not provide this field, the MasterPass Online displays as an option on the Payment Server card selection page.

MasterPass Online supports the following card types:

- Mastercard
- Amex
- Dinersclub (includes Discover)
- Visa

If your merchant profile is not configured for any of the supported card types for the transaction currency, then MasterPass Online does not display as an option on the Payment Server card selection page.

> **Note:** You must comply with the user interface branding requirements from MasterPass Online when you present MasterPass Online as an option to your customers on your website.

The following steps illustrate the payment flow in a 3-Party MasterPass Online transaction.

1	A customer browses your online store, selects one or more products and is ready to check out at the checkout page.
	a) The customer selects MasterPass Online on your website. If you indicate MASTERPASS as the Card Source in the redirect request to the Payment Server, the customer is directly sent to the MasterPass Online website.
	b) If the Card Source is not provided, the Payment Server redirects to the Payment Server card selection page and presents MasterPass Online as an option. When the customer selects MasterPass Online on the card selection page, the customer is redirected to the MasterPass Online website.

2	The customer logs into their MasterPass Online account and selects from the available payment and shipping options.
	If the merchant is configured to collect the shipping address from MasterPass Online, then MasterPass Online requests a shipping address from the customer. The customer may select a shipping address previously stored with MasterPass Online, or enter a new one. Alternatively, the merchant can collect the shipping address on his website before redirecting to MasterPass Online.
	Note 1: If you provide a billing or shipping address in the 3-Party payments request and also allow MasterPass Online to collect a billing or a shipping address, then the address returned by MasterPass Online will completely override the address provided in the 3-Party payments request.
	Note 2: The Payment Server does not allow recalculating the order amount using the shipping address selected by the customer at MasterPass Online.
3	The customer is redirected from MasterPass Online back to the merchant's website, via the Payment Server, with the transaction response. The response contains the result of the transaction including the masked card details, the billing address, and the shipping address (if applicable).
4	The merchant's website interprets the response and displays the receipt and confirms the order to the customer for their records.

Note: If MasterPass Online returns non-Latin-1 characters in the response then the Payment Server will reject the request with the exception of certain fields. For details on these fields, see MasterPass Online fields in the Virtual Payment Client Reference Guide.

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

In 3-Party Payments you can choose to bypass the Payment Server payments page that displays the logos of all the cards the Payment Provider will accept. This can be helpful if your application already allows the cardholder to select the card they want to pay with. This stops the cardholder having to do a double selection, once at your application and once on the Payment Server.

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

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

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

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

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

3-Party Payments using Verified by Visa™ and MasterCard SecureCode™

In 3-Party Payments you can have your Payment Provider enable you to use 3-D Secure which provides additional security to 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, see *Using 3-D Secure Payment Authentications* on page 15.

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.

5   Integrating 2.5 Party Payments

This information refers to the integration model for 3-Party payments with card details.

If you require the 2-Party Payments integration model and the 3-D Secure scheme, you will have to implement the 3-Party Payment with Card Details Integration Model, which is also referred to as 2.5 Party Payments.

In this model you are initialising a normal 3-Party Payment transaction but at the same time you, as the merchant, also collect and send the card details to the Payment Server. This results in the card selection and card details collection procedures to be bypassed, skipping directly to the 3-D Secure Authentication stage.

For more details see *Additional 3-Party Functionality* on page 29.

6   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.

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

Address Verification Service (AVS)

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

> **Note:** AVS is only supported for the S2I Acquirer. Furthermore, not all countries/banks support AVS, so even though AVS data is passed in the transaction data, if the issuing bank does not support AVS, it is ignored. Contact your acquiring bank to determine if AVS is applicable in your country.

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.

On MasterCard and Visa credit cards, the CSC is the three-digit number printed on the signature panel on the back.

On American Express cards, the CSC is the four-digit number printed on the front above the credit card account number.

In a 2-Party transaction and a 3-Party transaction 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. The level of the match between what the cardholder's 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 where 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's 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 log file.
>
> CSC is mandatory in some countries and regions. Check with your Payment Provider to determine the requirements of your region.
>
> Furthermore, not all banks support CSC, so even though CSC data is passed in the transaction data, if the issuing bank does not support CSC, it is ignored.

Card Present Transactions

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 or Purchase transaction.

> **Note:** Card Present transactions are not always available. Check with your Payment Provider to determine if card present transactions meet the requirements of your region.

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. This data is then included in the Transaction Request with an Authorisation or Purchase transaction. Card present functionality can only be performed as a 2-Party Authorisation/Purchase transaction.

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

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

With card track data,

- If both **vpc_CardTrack1** and **vpc_CardTrack2** are available, both must be added to the Transaction Request

or

- If only **vpc_CardTrack1** or **vpc_CardTrack2** is available, it 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.

External Payment Selection (EPS)

EPS is only used in a 3-Party transaction for bypassing the Payment Server page that displays the logos of all the cards the payment processor will accept. This can be helpful if the merchant's application already allows the cardholder to select the card they want to pay with. This stops the cardholder having to do a double selection - once at the merchant's application and once on the Payment Server.

Merchant Transaction Source

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

- INTERNET – indicates an Internet transaction
- MOTOCC – indicates a call centre transaction
- MOTO – indicates a mail order or telephone order
- MAILORDER – indicates a mail order transaction
- TELORDER – indicates a telephone order transaction
- CARDPRESENT – indicates that the merchant has sighted the card.

  This can only be used if the merchant has the privilege set to use this functionality, otherwise the transaction is 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.

> **Note**: If this field is not present the default value set in the Merchant ID Profile will be used.

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 the privilege set to use this functionality, otherwise the transaction is 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 account 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. The merchant is responsible for generating recurring payments.
>
> **Note:** If this field is not present, the default value set in the Merchant ID Profile will be used.

Enhanced Industry Data

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

- Ticket Number – allows the merchant to submit airline ticket number in the Transaction Request, including Capture transactions. The previous ticket number is overwritten when a new ticket number is submitted. The Payment Server does not maintain an audit record of these changes. You can view the latest Ticket Number in the search results of a Transaction Search using the Merchant Administration portal on the Payment Server.

- Addendum Data – allows the merchant to include industry specific data in the Transaction Request. The data can include passenger names, ticket numbers, hotel bookings, etc. The addendum data is stored in the database, which may be used in creating reports external to the Payment Server.

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

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

Risk Management

Risk Management is a module used for Card-Not-Present (CNP) transactions, which enables MSOs and merchants to use a set range of business risk screening rules. These risk rules are configured to screen transactions of perceived high or low risk thereby enabling merchants to accept, reject, or mark transactions for review based on their risk assessment.

The solution introduces various rules for risk mitigation where each rule contributes differently to the risk profile. IP Address Range and Card BIN rules enable blocking transactions from high-risk IP Address ranges and high-risk BIN ranges respectively. Trusted Cards and Suspect Cards allow you to create lists of trustworthy card numbers and suspect card numbers respectively. 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, trusted cards and AVS rules can be configured at the merchant level only.

MSOs have the added privilege 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 portals, 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 portals on the Payment Server.

> **Note:** Merchants must have the **May Use Risk Management** privilege enabled to use Risk Management. To enable this feature contact your MSO.

> **Note:** Risk Management is applicable only to **Auth Then Capture** and **Purchase** transaction modes. Standalone Captures, Standalone Refunds, etc., will not be assessed for risk.

The diagrams below illustrate the process flow of an order when risk management is enabled for an MSO and merchant respectively.
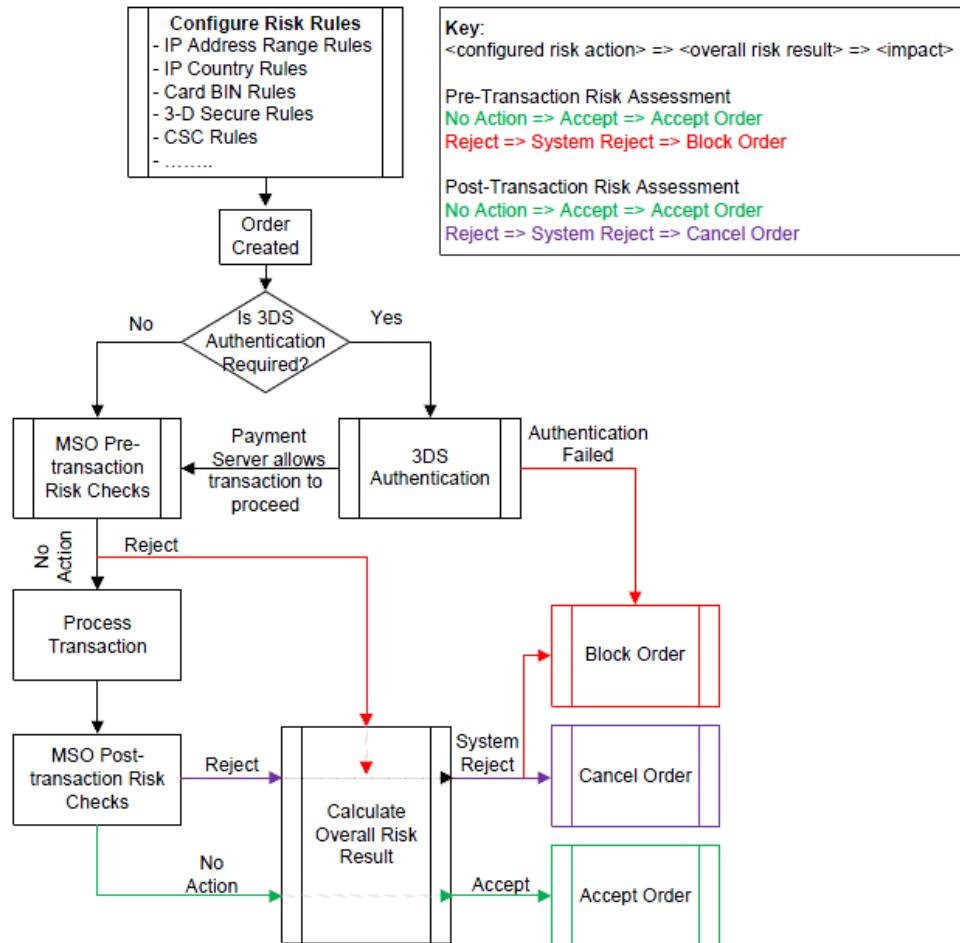
Diagram 1 Process Flow for an MSO when Risk Management is Enabled
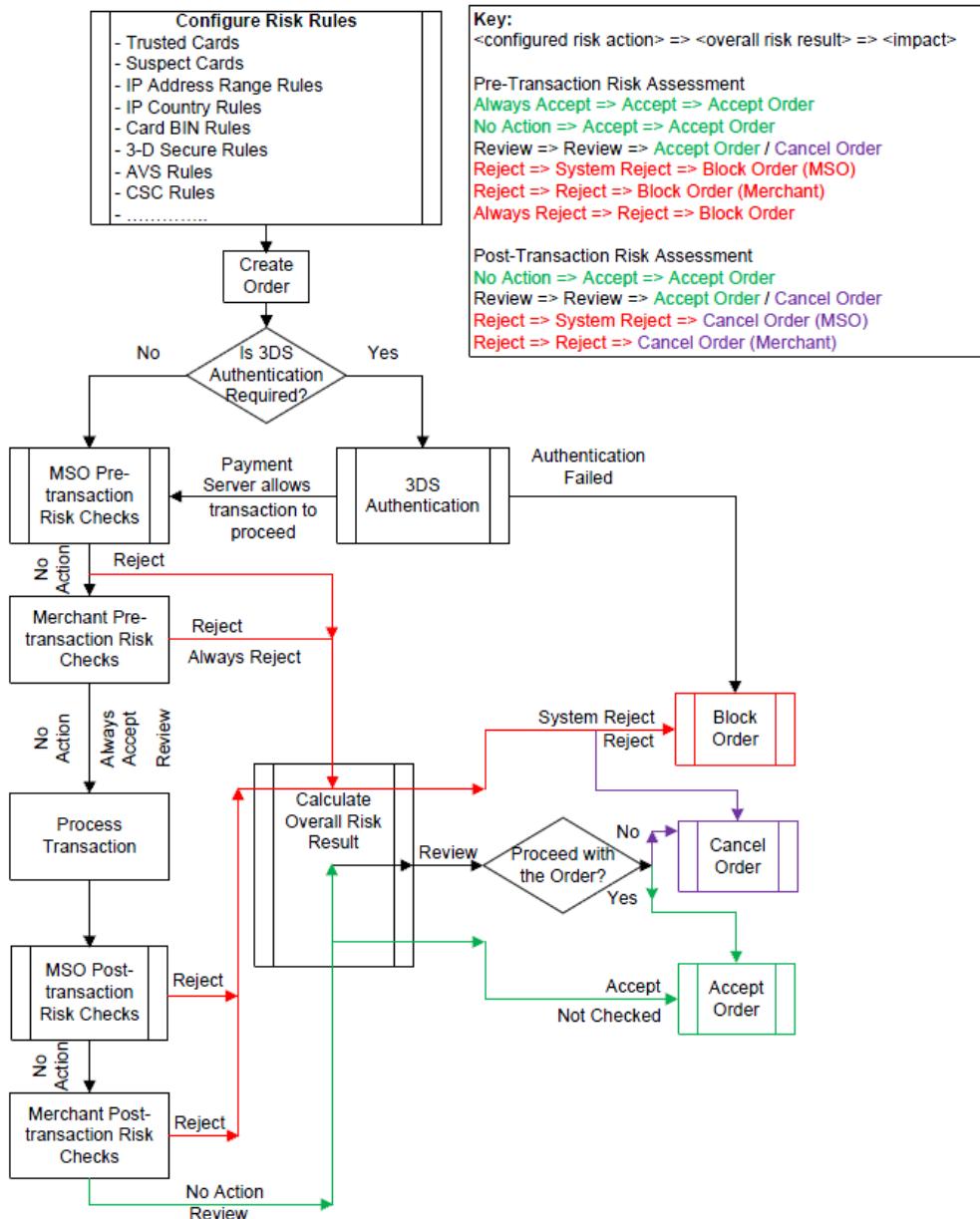
Diagram 2 Process Flow for a Merchant when Risk Management is Enabled

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 passed to the acquirer in the settlement file.

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

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.

The cardholder selects the account type (Savings or Cheque) to use for the transaction. For 2-Party transactions, the merchant sends the account type with the Transaction Request. For 3-Party transactions, the cardholder selects it on the 3-Party payment page.

The Account Type is stored on the Payment Server, and will be displayed in the Order Search results in the Merchant Administration portal.

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

EMV Integrated Circuit Card (EMVICC)

The EMVICC card field is applicable to MasterCard and Visa cards that have an embedded integrated circuit chip, providing a greater level of security and fraud prevention. If this is supported for a merchant, they can select to capture this data in a transaction. It is stored in the Payment Server database with other order details.

Note: Applies to Authorisation, Capture and Purchase transactions only.

Note: The EMVICC data is not displayed in either the Merchant Manager or Merchant Administrator.

7 Payment Authentication

MasterCard® SecureCode™, Verified by Visa™, JCB J/Secure™, American Express SafeKey™ and Diners Club ProtectBuy™ are 3-D Secure (3DS) payment authentication methods designed to reduce credit card fraud by authenticating cardholders when performing transactions over the Internet.

A payment authentication is performed immediately before a merchant performs an authorisation or purchase. Authenticating ensures that the card is being used by its legitimate owner.

During a transaction, authentication allows a merchant to confirm the identity of the cardholder by redirecting them to their card issuer where they enter a password that they had previously registered with their card issuer.

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 for 3DS with the appropriate card scheme.

Payment Authentication 3-D Secure transaction modes

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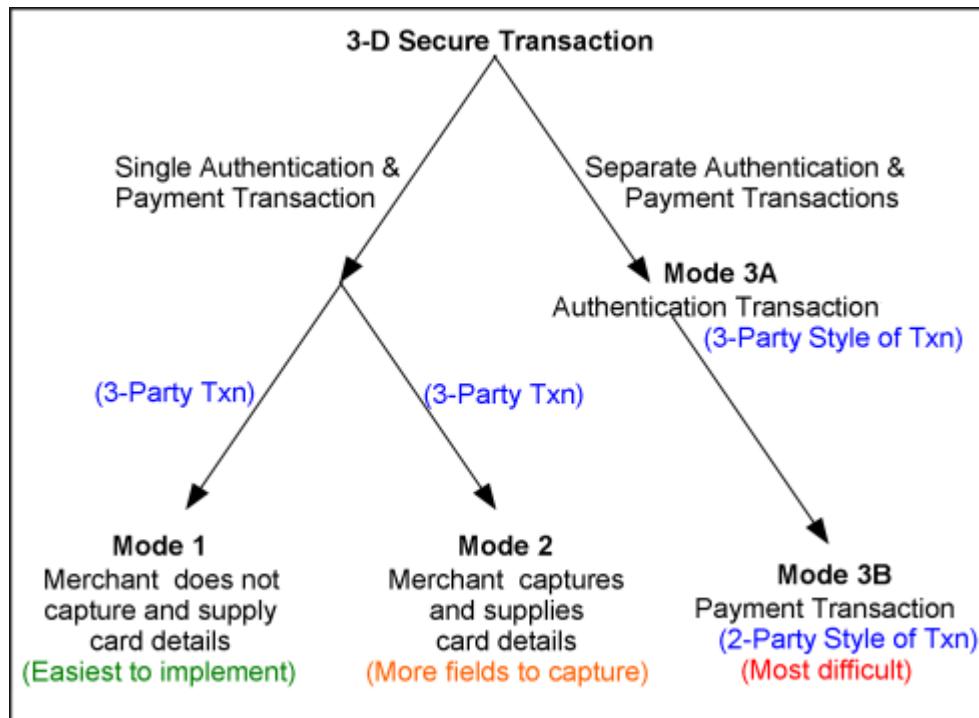
Diagram 3 3-D Secure Transaction Modes

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

    The Payment Server, and not the merchant's application, collects the cardholder's card details. 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.

Table 1 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)
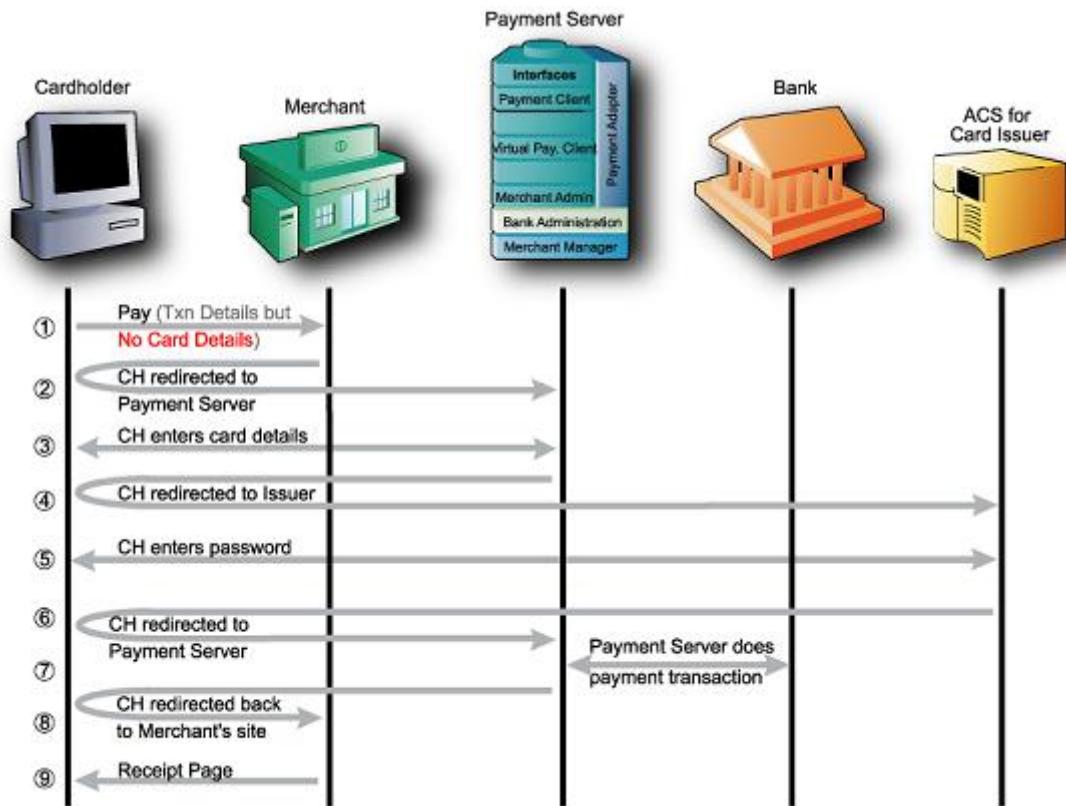

The information flow for a successful 3-Party Authentication and Payment transaction mode 1 is very similar to a standard 3-Party transaction.

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 (step 8) with a **vpc_TxnResponseCode** of 'F'. No payment takes place in this scenario.
7. If the cardholder is authenticated correctly, the Payment Server continues with processing the payment part of the transaction. The cardholder is redirected back to the merchant, where the receipt is passed back to the merchant.
8. The merchant displays a receipt page to the cardholder to indicate whether the transaction was successful or not.

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

The information flow for this transaction where the merchant collects the card details uses the basic 3-Party style of transaction with some additional input fields. For more information, refer to basic 3-Party Transaction.
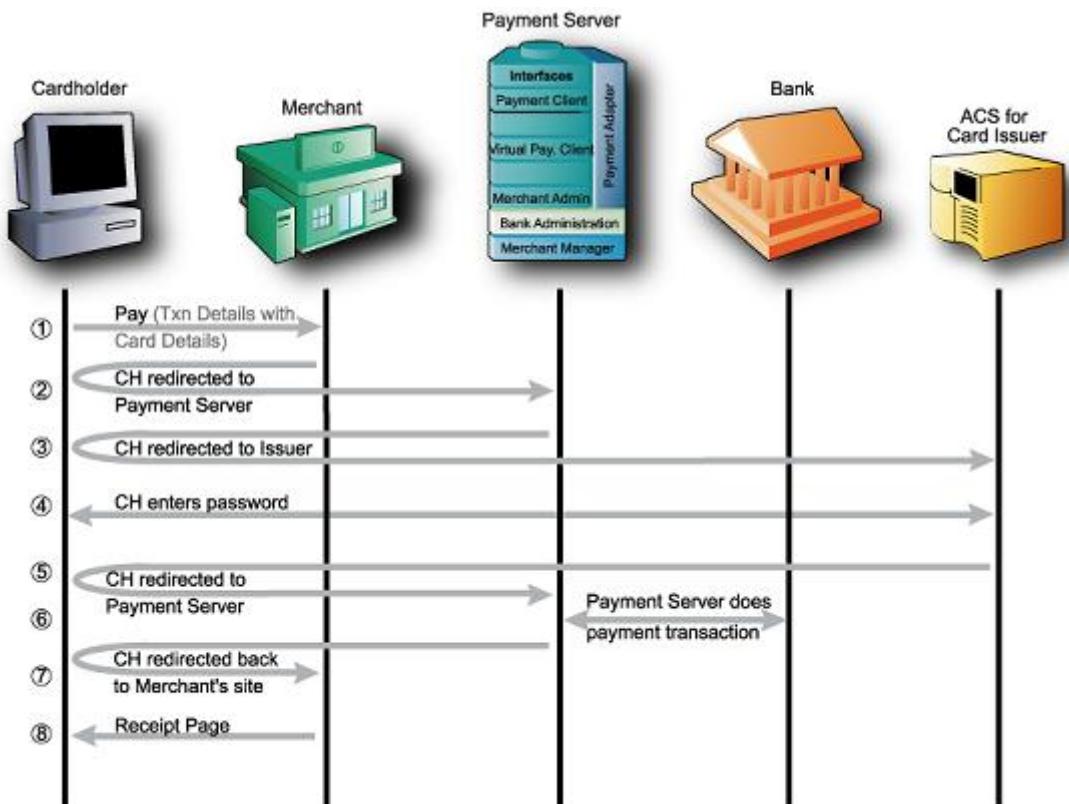
The information flow for a 3-Party Authentication and Payment mode 2 transaction is:

1. The cardholder enters their card details into the merchant's application and clicks the merchant's Pay button.

2. Their browser is redirected to the Payment Server and the Payment Server determines if the card is enrolled in 3-D Secure for the appropriate card scheme.

   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, and 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 (step 7) with a **vpc_TxnResponseCode** of '**F**'. No payment takes place in this scenario.

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

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

8. The merchant displays a receipt page to the cardholder to indicate whether the transaction was successful or not.

Mode 3a – Implementing a 3-Party Style Authentication Only transaction

In a 3-Party Authentication Only transaction mode 3a, 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 where a merchant may want to perform an immediate authentication, but delay the payment transaction.

The information flow for this transaction mode uses the 3-Party style of transaction with additional card details. For more information, refer to basic 3-Party Transaction.
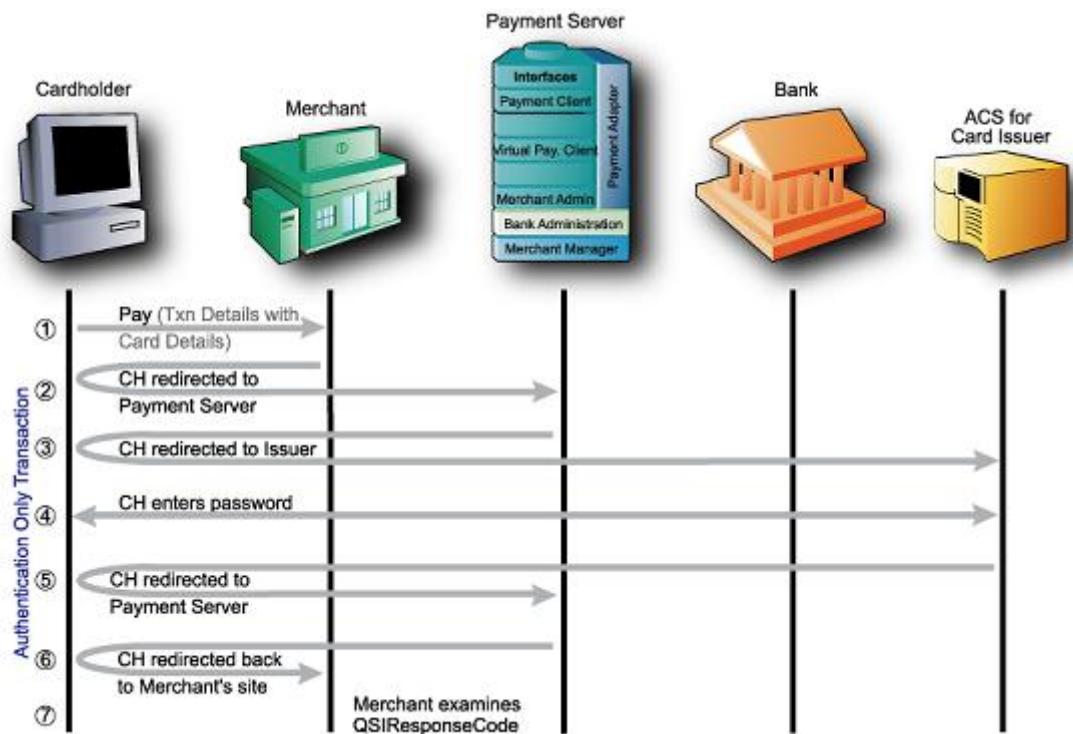
The information flow for a 3-Party Authentication Only mode 3a transaction is:

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 3-D Secure for the appropriate card scheme.

    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, and 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 Pre-authenticated Payment transaction uses the 2-Party style of transaction. For more information on 2-Party transactions, see *2-Party Payments Integration Model* on page 5).
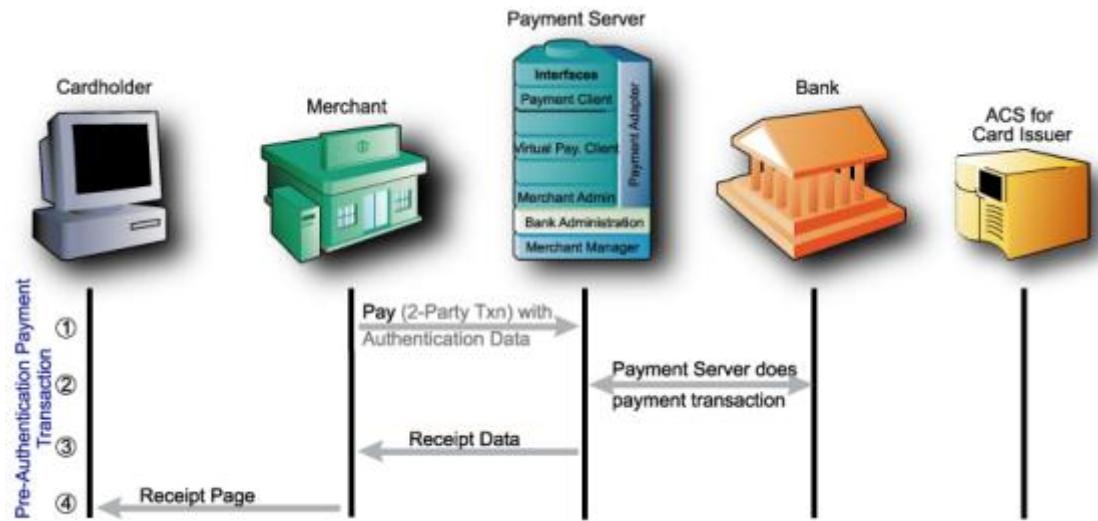
The information flow for this transaction is:

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 to indicate whether the transaction was successful or not.

8   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 perform a particular AMA transaction.

Capture

The AMA Capture command allows you 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 account.

2. The second transaction (Capture) transfers the funds from the cardholder's account to the merchant's account.

Capture allows you 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 your account.

Note: In Purchase mode, the authorisation and capture operations are completed at the same time in the one purchase transaction, so a separate capture is not needed. This Capture command is not necessary if you are 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 in 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 your account back to the cardholder's account.

Refunds can only be performed for a previously completed purchase or capture transaction for the particular order. You can perform 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 you do not have many Refund transactions. For more information, refer to your Merchant Administration User Guide.

2. Using the AMA Refund command in the Virtual Payment Client to directly perform refunds from your application.

Standalone Refund

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

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

Void Capture

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

This command cannot be used if you are operating in Purchase mode.

You can only perform 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 performed before the batch containing the original capture transaction is processed by the acquiring institution.

There are two ways you can perform a Void Capture. 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 Capture transactions. For more information, refer to your Merchant Administration User Guide.

2. Using the Void Capture command in the Virtual Payment Client to directly perform Void Captures from your application. You must have a user enabled with AMA and Void privileges to use this functionality.

Note: Check with your financial institution if it supports void captures.

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

Void Refund

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

You can only perform one Void Refund transaction on the original refund transaction as it completely removes the refund transaction as though it never occurred.

A Void Refund must be performed before the batch containing the original refund transaction is processed by the acquiring institution.

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 in the Virtual Payment Client to directly perform Void Refunds from your application. You must have a user enabled with AMA and Void privileges to use this functionality.

Note: Check with your financial institution if it supports void refunds.

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

Void Purchase

AMA Void Purchase allows you to void a previous 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.

You can only perform one Void Purchase transaction on the original purchase transaction as it completely removes the purchase transaction as though it never occurred.

A Void Purchase must be performed before the batch containing the original purchase transaction is processed by the acquiring institution.

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 refer to your Merchant Administration User Guide.

2. Using the Void Purchase command in the Virtual Payment Client to directly perform Void Purchases from your application. You must have a user enabled with AMA and Void privileges to use this functionality.

Note: Check with your financial institution if it supports void purchases.

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

Excessive Captures

The Excessive Captures feature enables you to perform captures for amounts greater than the authorised amount. This feature is available for orders performed in Merchant Administration or through the 2-Party gateway.

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

QueryDR

The AMA QueryDR command allows you to search for a duplicate transaction receipt. The search is performed on the key **vpc_MerchTxnRef**, so the **vpc_MerchTxnRef** field must contain a unique value.

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

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

QueryDR always returns these two 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.

Note: QueryDR does not return receipt data for 3-D Secure Authentication Only transactions.

Troubleshooting

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

What if a Session Timeout occurs?

The current 3-Party Payment Timeout value is set at 15 minutes.

It is possible that while a cardholder is entering their card details at the Payment Server, the session is terminated (e.g. by 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** 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 merchant ID or password is set incorrectly for a merchant profile in the Payment Server Merchant Manager.

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 transactions.

9 Glossary

Term	Description
Access Code	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 transaction that uses the information from an authorisation 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 bank or financial institution that issues credit cards to customers.
Merchant Administration	Allows you to monitor and manage your electronic transactions through a series of easy to use, secure web pages.
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	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	A single transaction that immediately debits the funds from a cardholder's credit card account.
RRN	The 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	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	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 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.

10   Appendix A

Test Environment – Test Cards

The following table shows the test card numbers and associated expiry dates configured for each card scheme on the MiGS Payment Server.

Table 2 Test Cards

Card Type	PAN	Expiry
MasterCard	5123456789012346	05/17
MasterCard	5313581000123430	05/17
VISA	4005550000000001	05/17
VISA	4557012345678902	05/17
AMEX	371449635311004	05/17
Bankcard (Australian Domestic)	5610901234567899	05/17
Diners	30123456789019	05/17