# Submitting Transactions in the Direct Model Using 3D Secure Authentication Guide

**Version 5.0 – August 2013**

**Business Gateway**

This page is intentionally blank.

# Table Of Contents

# About this Guide

This guide describes the specifications of XML orders sent to the WorldPay system in the XML Direct model. The intended audience is the merchant's technical staff or the merchant's system integrator.

Because almost all communication between the merchant's system and the WorldPay Payment Service is realised through predefined XML messages over the Internet using standard protocols, you will need basic XML programming skills and knowledge of HTTP(S). Furthermore, it is recommended that you are familiar with the basics of the Payment system. Where applicable, this document refers to the related documentation with further details.

## *Update History*

| Version | Change description | Date |
|---|---|---|
| 5.0 | Added information about V.me by Visa digital wallet service. | August 2013 |
| 4.9 | Test card numbers and payment method codes have been updated. | March 2013 |
| 4.8 | Error Fixes. | January 2013 |
| 4.7 | Information about alternative payment methods (supported by WorldPay AP Ltd.) has been moved to the Alternative Payment Methods Guide. | December 2012 |
| 4.6 | • Updated the list of alternative payment methods and the maximum and minimum amounts.<br>• Added information about mandatory and optional fields for alternative payment methods.<br>• Added information about transaction statuses returned in pendingURL. | September 2012 |
| 4.5 | • Updated the list of alternative payment methods.<br>• Added the maximum and minimum amounts for alternative payment methods. | July 2012 |
| 4.4 | Added code examples for alternative payment methods. | June 2012 |
| 4.3 | Payment method code for Yandex.Money corrected. | May 2012 |
| 4.2 | The XML code for an order submission has been updated. | April 2012 |
| 4.1 | Updated link to the latest DTD.<br>Added the statementNarrative element. | March 2012 |
| 4.0 | Gateway and guide name added to navigation path. | December 2011 |

| | | |
|---|---|---|
| 3.1 | New cardholder authentication response. | October 2011 |
| 3.1 | Payment page update. | October 2011 |
| 3.0 | WorldPay rebrand. | July 2011 |

# *Copyright*

# Introduction

XML (Extensible Markup Language) is a set of rules for encoding documents electronically and is a universal way of exchanging documents and data across applications and platforms. It is used by WorldPay to send pre-defined messages containing technical information about a payment between our own payment service and a merchant's system.

Our payments service allows merchants who use the XML Direct model to:

- submit orders with payment details
- send a modification for the order
- perform an inquiry to request the payment status of the order

## *What is the XML Direct Model?*

Merchants who collect and store their shoppers' payment details on their own platform can use the XML Direct model as an effective payment-processing gateway. With this model, the merchant collects both order and payment details and then communicates the relevant payment details on a per order basis with WorldPay, for processing.

The benefits of using the Direct Method for the merchant include being able to retain full control over the payment process and also the payment pages displayed to shoppers.

However, for this method to work successfully, it is important that the merchant ensures their own system operates within a secure environment so that payment details, which they collect and store, are protected. In view of the cost involved in establishing appropriate security measures, this model only applies to merchants with established high transaction volumes.

## Security

A core issue associated with using the XML Direct model is security. The collection and storage of payment information such as card numbers and cardholder names must take place in a secure environment.

### Payment Card Industry Data Security Standard (PCI DSS)

PCI DSS is a global Card Scheme initiative that aims to ensure that every entity that handles, stores or processes cardholder data does so in a secure manner. MasterCard and Visa have combined their own security standards for cardholder data creating an aligned program, which is now endorsed by American Express, JCB and Diners. Much of PCI DSS relates to the technology involved in capturing and processing card data and this is particularly relevant to those merchants who process and capture cardholder data on their own systems rather than those who use the secure WorldPay payment pages.

For more information, please refer to PCI DSS and to the PCI Security Standards Council at: www.pcisecuritystandards.org. If you want any help to gain compliance this site also lists PCI approved Quality Security Assessors (QSA) who can provide technical advice (chargeable). WorldPay does not take responsibility for an external link's operation or content.

## 3-D Secure Authentication

You can reduce your exposure to fraud and increase confidence in online shopping by implementing 3-D Secure Authentication with the XML Direct Model. The additional security benefits and liability shifts of authenticated transactions are currently only supported by Visa, MasterCard, and American Express SafeKey.

*MasterCard requires that your website implements 3-D Secure authentication in order for you to continue accepting Maestro payments.*

The benefits of the 3-D Secure process are the enhanced security available when performing an authenticated transaction as well as the shift of liability in the event of fraudulent transactions. Authentication should strengthen your existing anti-fraud strategy and help protect your business, but bear in mind that coverage of authentication programmes is currently limited to Internet transactions. This means that authentication programmes do not cover fax, mail, or phone orders, nor do they cover all card types.

For further details about authentication, refer to the Cardholder Authentication guide

# *Introduction*

XML (Extensible Markup Language) is a set of rules for encoding documents electronically and is a universal way of exchanging documents and data across applications and platforms. It is used by WorldPay to send pre-defined messages containing technical information about a payment between our own payment service and a merchant's system.

Our payments service allows merchants who use the XML Direct model to:

- submit orders with payment details
- send a modification for the order
- perform an inquiry to request the payment status of the order

# *What is the XML Direct Model?*

Merchants who collect and store their shoppers' payment details on their own platform can use the XML Direct model as an effective payment-processing gateway. With this model, the merchant collects both order and payment details and then communicates the relevant payment details on a per order basis with WorldPay, for processing.

The benefits of using the Direct Method for the merchant include being able to retain full control over the payment process and also the payment pages displayed to shoppers.

However, for this method to work successfully, it is important that the merchant ensures their own system operates within a secure environment so that payment details, which they collect and store, are protected. In view of the cost involved in establishing appropriate security measures, this model only applies to merchants with established high transaction volumes.

Please also note that the XML Direct model only allows for using a select number of online payment methods, where no consumer interaction is involved.

## Security

A core issue associated with using the XML Direct model is security. The collection and storage of payment information such as card numbers and cardholder names must take place in a secure environment.

### Payment Card Industry Data Security Standard (PCI DSS)

PCI DSS is a global Card Scheme initiative that aims to ensure that every entity that handles, stores or processes cardholder data does so in a secure manner. MasterCard and Visa have combined their own security standards for cardholder data creating an aligned program, which is now endorsed by American Express, JCB and Diners. Much of PCI DSS relates to the technology involved in capturing and processing card data and this is particularly relevant to those merchants who process and capture cardholder data on their own systems rather than those who use the secure WorldPay payment pages.

For more information, please refer to PCI DSS and to the PCI Security Standards Council at: www.pcisecuritystandards.org. If you want any help to gain compliance this site also lists PCI approved Quality Security Assessors (QSA) who can provide technical advice (chargeable). WorldPay does not take responsibility for an external link's operation or content.

## 3-D Secure Authentication

You can reduce your exposure to fraud and increase confidence in online shopping by implementing 3-D Secure Authentication with the XML Direct Model. The additional security benefits and liability shifts of authenticated transactions are currently only supported by Visa, MasterCard, and American Express SafeKey.

*MasterCard requires that your website implements 3-D Secure authentication in order for you to continue accepting Maestro payments.*

The benefits of the 3-D Secure process are the enhanced security available when performing an authenticated transaction as well as the shift of liability in the event of fraudulent transactions. Authentication should strengthen your existing anti-fraud strategy and help protect your business, but bear in mind that coverage of authentication programmes is currently limited to Internet transactions. This means that authentication programmes do not cover fax, mail, or phone orders, nor do they cover all card types.

For further details about authentication, refer to the Cardholder Authentication guide

This page is intentionally blank.

# Before You Start

## *Overview of Setting a Connection*

From a technical / IT viewpoint, setting up a successful connection between your system and the WorldPay payment system, involves the following major steps:

- *order creation*: automating the creation of an XML document that conforms with the rules specified in the WorldPay Document Type Definition (DTD), from order data captured from your shoppers by your systems. This topic is covered in detail in this guide.

*Order creation can be achieved in a number of different ways depending on the scripting language that you are using. This guide provides an overview only. WorldPay cannot advise on how to specifically create an XML order using ASP or PHP, or how to establish the secure connection*

- *account settings*: before sending test orders to the Payment Service check and, if necessary, update settings that control aspects of how the service works. For example, setting a delay between authorisation and capture.
- *order submission*: setting up an HTTP(S) connection between your system and ours and automating the sending of XML orders over that connection. This topic is covered in this guide.

### Connect Using Login Name and XML Password

When setting up the HTTP(s) connection over which you will send XML orders, remember to use a valid login and password.

Your login name is the relevant merchant code (to look up the merchant code(s) allocated to your organisation, login to the Merchant Interface and refer to the status box in the left-hand side of the page).

## *Error Messages, Invalid XML*

The most common cause of error when posting XML to the Payment Service is incorrectly formed XML.

Most of the incorrect XML issues will be resolved by using an industry standard parser. XML messages should be parsed prior to sending them to the WorldPay Payment Service and upon receipt. This ensures that the messages sent are valid and messages received are correctly interpreted. When parsing, please ensure that you are using an industry standard Parser that parses the message against the WorldPay Document Type Definition (DTD) (refer to Reference the DTD below). Do not rely on a self-built parser. Please refer to http://www.xml.org for further details on XML and parsers.

For examples of XML errors messages our system may return, please refer to the appendix XML Error Codes.

XML messages must adhere to the following rules in order to be accepted.

## Use Valid Syntax

All XML messages sent to the WorldPay system must be well-formed and valid. For the XML to be well-formed, it must adhere to a number of rules, including:

- every start tag must have a matching end tag
- elements must not overlap
- there must be exactly one root element
- attribute values must be quoted
- an element may not have two attributes with the same name
- comments and processing instructions may not appear inside tags
- no unescaped [<] or [&] signs may occur in the element's or attribute's character data.

## Reference the DTD

A valid XML message must always include a reference to the DTD, so it can be automatically compared with it in order to check if all the references used are correct.

The DTD is specified in the header of the XML message and will look as follows:

```
<?xml version="1.0"? encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay
PaymentService v1//EN" "http://dtd.worldpay.com/paymentService_v1.dtd">
```

## Use Declared Elements Only

Every element, attribute and entity in the XML that you send, must be declared in the DTD (Document Type Definition).

XML elements can be declared to contain:

- name tokens (NMTOKEN)
- parsed character data (PCDATA)
- character data (CDATA) or constants.

The DTD can be found at: http://dtd.worldpay.com/paymentService_v1.dtd

## Name Tokens

An XML name token consists of alphanumeric and/or ideographic characters and the punctuation marks [_], [-], [.], and [:]. No other characters are allowed. An XML name token may not contain white spaces.

If an attribute is declared to contain a name token or list of name tokens, the values of these attributes must be legal XML name tokens. Below you can find an example of this:

```
<!ELEMENT amount EMPTY>
<!ATTLIST amount value NMTOKEN #REQUIRED
currencyCode NMTOKEN #REQUIRED
exponent (0 | 2 | 3 ) #REQUIRED
debitCreditIndicator (debit | credit ) 'credit' >
```

A valid instance of an amount element looks like this:

```
<amount value="4938" currencyCode="SEK" exponent="2" />
```

## PCDATA

In a PCDATA (Parsed Character DATA) block in the XML message the following special characters should not be included: [&], [<], [>] and ["]. If you need to use these characters within a PCDATA block you should specify them as follows:

& = &

> = >

< = <

" = "

For example, for the description element which is declared to contain PCDATA <!ELEMENT description (#PCDATA )> a valid example would be:

```
<description>Your Holland &amp; Holland Catalogue</description>
```

## CDATA

In a CDATA (Character DATA) block it is allowed to include any data / characters as long as it adheres to the specified encoding and does not contain the character sequence that follows, which is the character set used to determine the end tag:

```
]]>
```

A CDATA block must be enclosed between the start tag <[CDATA[ and the end tag ]]>. For example:

```
<[CDATA[This text has not been parsed & still can be used]]>
```

# Structure of an XML Direct Order

## *Introduction*

Orders submitted to the WorldPay system are required to be valid XML files as specified in this guide and in the Document Type Definition (DTD), available at:

```
http://dtd.worldpay.com/v1/
```

XML files are valid if they are well-formed, that is, they have a correct XML syntax and conform to a Document Type Definition. The content of the XML orders should always be in compliance with your contract with WorldPay and should not exceed 4k in size.

## *XML and Document Type Declaration*

As with all well-formed valid XML documents, an XML Direct order submission begins with an XML declaration and a document type declaration, containing the root element `paymentService` and the reference to our public DTD:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.worldpay.com/paymentService_v1.dtd">
```

## *Order Description and Amount*

Within the `submit` element, the `order` element and its content describe the goods or services that are being ordered.

The `order` element has two required attributes:

- An `orderCode` attribute whose value must be *unique*. Order codes can be up to 64 characters long; neither spaces, quotes nor the "<" and ">" characters are allowed.

  You may use the `orderCode` attribute to contain a Cart ID as long as it is unique. If a Cart ID is not unique, then we recommend you use either:

  - the `description` element (see below) to enter the Cart ID; or
  - use the `orderCode` attribute but append a unique number to a static Cart ID.

*An order with a previously used order code cannot be processed correctly and you are likely to receive error messages and encounter problems with reporting.*

- An `installation id` attribute. This should be your XML Invisible installation id that you have received from WorldPay.

The first two child elements of the `order` element are `description` and `amount`. The `description` element should contain a simple one-line description of the order and can be

up to 255 characters long. The `amount` element has the attributes: `value` (no decimal point or comma), the `currencyCode` (ISO 4217 code) and `exponent` (specifies where the decimal point or comma should be placed, counting from the right). The amount value is the total amount the shopper is expected to pay. A list of currency codes and their respective exponents can be found in the appendix ISO Currency Codes.

```
<order orderCode="T0211010" installationId="12345">
  <description>20 English Roses from MYMERCHANT Webshops</description>
  <amount currencyCode="GBP" exponent="2" value="5000"/>
  ...
</order>
```

## Order Content

The third child element of the `order` element is `orderContent`. You can deliver the order content in HTML format. When supplying HTML order content the *only* HTML tags allowed are the tags permitted between the `<body>` and `</body>` tags of a valid HTML document! No form of scripting is allowed in the order content.

The order content must be less than 10 kilobytes and should always be included in a `CDATA` section to avoid parsing problems.

```
<orderContent>
  <![CDATA[content here]]>
</orderContent>
```

## Payment Details and Session Information

The fifth `order` child element is `paymentDetails` and this contains the details of the selected payment method. Each payment method has its own set of sub-elements and attributes. Please refer to the specifications of the `paymentDetails` element in the DTD for an up-to-date list of available payment method codes for the Direct model and their child elements. The payment method codes are listed in the appendix Payment Method Codes.

The element `paymentDetails` must also include information that is used by our payment service to submit a 3-D Secure transaction successfully. This includes information that you must provide about the shopper's browser session in the child element `session`, containing the `shopperIPAddress` and `session ID`.

*WorldPay uses the payment details and session information for risk assessment. They are a mandatory element in a 3-D Secure transaction.*

The following is an example of the `paymentDetails` for a VISA payment, where `VISA-SSL` is the payment method code:

```
<paymentDetails>
    <VISA-SSL>
        <cardNumber>4444333322221111</cardNumber>
        <expiryDate>
            <date month="09" year="2009"/>
        </expiryDate>
        <cardHolderName>J. Shopper</cardHolderName>
        <cvc>123</cvc>
        <cardAddress>
```

```
            <address>
                <street>47A Queensbridge Rd</street>
                <postalCode>CB94BQ</postalCode>
                <countryCode>GB</countryCode>
            </address>
        </cardAddress>
    </VISA-SSL>
    <session shopperIPAddress="123.123.123.123" id="0215ui8ib1" />
</paymentDetails>
```

*Note that the payment method code VISA-SSL should be used for both Visa credit and Visa debit card payments.*

*Note that the `cvc` element contains the Card Verification Code. For details please refer to the appendix CVC/CVV Checks And Responses. Also note that the numeric parts (if there are any) of the `street` element are used in the AVS check. The non-numeric parts are ignored.*

The following example shows the `paymentDetails` for a V.me payment, where `VME-SSL` is the payment method code:

```
<paymentDetails>
    <VME-SSL>
        <successURL>http://worldpay.com/vme/vme_success.html</successURL>
        <failureURL>http://worldpay.com/vme/vme_failure.html</failureURL>
        <cancelURL>http://worldpay.com/vme/vme_cancel.html</cancelURL>
    </VME-SSL>
</paymentDetails>
```

*The V.me XML Direct payment method is implemented differently to most XML payment methods. For more information see: Submitting a V.me Order.*

The following example shows `paymentDetails` for the German payment method ELV-SSL:

```
<paymentDetails>
<ELV-SSL>
<accountHolderName>Johannes Kaufer</accountHolderName>
    <bankAccountNr>1234567</bankAccountNr>
    <bankName>My bank</bankName>
    <bankLocation>Berlin</bankLocation>
    <bankLocationId>12345678</bankLocationId>
</ELV-SSL>
</paymentDetails>
```

## Shopper Information

The `sixth` order child element is `shopper` and this contains further details about the cardholder making the purchase.

It includes the `shopperEmailAddress` element which is used by our payment service to identify possible fraudulent transactions and / or to send an email to the shopper when the payment is authorised or refused.

In addition, your system must also include details about the shoppers's browser settings using the elements: `browser`, `acceptHeader` and `userAgentHeader`. These must not be hard coded as this information is required to redirect the shopper to the correct issuer site for 3-D Secure authentication. Please note that the shopper will only be redirected for authentication if it is verified by WorldPay that they have enrolled with the 3-D Secure scheme. The example below uses Firefox 3.5.5 to demonstrate.

```
<shopper>
    <shopperEmailAddress>jshopper@myprovider.int</shopperEmailAddress>
    <browser>
  <acceptHeader>text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</a
cceptHeader>
         <userAgentHeader>Mozilla/5.0 (Windows; U; Windows NT 5.1; en-GB;
rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5 (.NET CLR 3.5.30729)</userAgentHeader>
     </browser>
</shopper>
```

The `acceptHeader` element should contain the exact content of the HTTP accept header as sent to the merchant from the shopper's user agent.

The `userAgentHeader` element should contain the exact content of the HTTP user-agent header as sent to the merchant from the shopper's user agent.

## Billing Address

A seventh `order` child element is `shippingAddress`. It is an optional element that enables you to pre-populate the billing address fields that are part of the Payment Page we present to your customers when they indicate that their billing address is the same as the shipping address they have already supplied you.

Typically this would be done by offering a button to the shopper on your web pages to indicate that the billing and shipping address are the same and, hence, add the convenience that they don't have to key the address twice unless the addresses are different. You would then populate the `shippingAddress` element with the supplied shipping address and when you send it to the payment service its content is used to pre-populate the billing address fields on the Payment Page.

## Statement Narrative

Use the `statementNarrative` element to specify text that can be displayed on the shopper's statement. The `statementNarrative` element is the twelfth `order` child element.

*This element is currently supported by the Qiwi, AliPay and China Union Pay (CUP) payment methods.*

```
<statementNarrative>STATEMENT NARRATIVE TEXT</statementNarrative>
```

## XML Validation

When creating XML documents it is good practice to check the syntax of the candidate XML document and determine whether it conforms to its schema, expressed in the DTD. We

strongly recommend that you validate the XML your system creates before submitting it to the Payment Service. XML that does not conform to the WorldPay DTD is not accepted.

Numerous on-line and off-line tools are available to help you check and validate XML. For example, please refer to: http://xml.coverpages.org/check-xml.html.

## *XML Direct Order Examples*

An example of a complete XML order for the Direct model is shown below. The order is for 20 English roses and has an order code: T0211010. The merchant is the MYMERCHANT Webshop with merchant code MYMERCHANT, the shopper is Mr. J. Shopper and the used payment method is VISA.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="MYMERCHANT">
   <submit>
       <order orderCode="T0211010" installationId="12345">
             <description>20 English roses from MYMERCHANT Webshop</description>
             <amount value="2600" currencyCode="EUR" exponent="2"/>
             <paymentDetails>
                 <VISA-SSL>
                     <cardNumber>4444333322221111</cardNumber>
                     <expiryDate>
                         <date month="09" year="2019"/>
                     </expiryDate>
                     <cardHolderName>J. Shopper</cardHolderName>
                     <cvc>123</cvc>
                     <cardAddress>
                         <address>
                              <street>47A Queensbridge Rd</street>
                              <postalCode>CB94BQ</postalCode>
                              <countryCode>GB</countryCode>
                         </address>
                     </cardAddress>
                  </VISA-SSL>
                  <session shopperIPAddress="100.100.100.100" id="0215ui8ib1" />
             </paymentDetails>
             <shopper>

 <shopperEmailAddress>jshopper@myprovider.int</shopperEmailAddress>
                 <browser>

 <acceptHeader>text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8</ac
ceptHeader>
                      <userAgentHeader>Mozilla/5.0 (Windows; U; Windows NT 5.1;
en-GB; rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5 (.NET CLR
3.5.30729)</userAgentHeader>
                  </browser>
              </shopper>
       </order>
  </submit>
</paymentService>
```

## V.me XML Order Example

The following example shows an XML submission containing all `paymentDetails` for a V.me payment:

```
 <?xml version="1.0" encoding="UTF-8"?>
 <!DOCTYPE paymentService PUBLIC "-//WorldPay/DTD WorldPay PaymentService v1//EN"
http://dtd.worldpay.com/paymentService_v1.dtd">


 <paymentService version="1.4" merchantCode="MYMERCHANT">
  <submit>
   <order orderCode="vmeexampleorder123" shopperLanguageCode="en">
    <description>V.me test order</description>
    <amount value="100" currencyCode="EUR" exponent="2"/>
    <orderContent>
     <![CDATA[]]>
    </orderContent>
    <paymentDetails>
     <VME-SSL>
      <successURL>http://worldpay.com/vme/vme_success.html</successURL>
      <failureURL>http://worldpay.com/vme/vme_failure.html</failureURL>
      <cancelURL>http://worldpay.com/vme/vme_cancel.html</cancelURL>
     </VME-SSL>
    </paymentDetails>
    <shopper>
     <shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>
    </shopper>
   </order>
  </submit>
 </paymentService>
```

# Response from the Payment Service

## *Introduction*

When the Payment Service has received a valid order with payment details it will send the information to the financial institutions (acquirers) for authorisation. The result of the authorisation request is reported to WorldPay online as either AUTHORISED or REFUSED (or 'ERROR' if there is a temporary problem with the order submitted). In the XML Direct model, WorldPay sends an XML response to your system that contains the payment status of the order. For further information about the different payment statuses reported, please refer to Acquirer Response Codes.

*When parsing WorldPay reply it is important that you use an industry standard XML parser. Do not depend on a homemade one, which may not be able to correctly interpret the messages received from WorldPay. For various platforms different XML parsers exist. Please refer to http://www.xml.org.*

The topics covered in this chapter are listed below.



## *Reply to an XML Direct Order*

The following provides some examples of replies sent by our payment service in response to an XML Direct Order.

### Example of an Authorised Response

An AUTHORISED reply is sent when the financial institution has approved the payment. Please note that while an AUTHORISED reply is usually an indication that the card details submitted are valid, it is not a guarantee of payment. For further information about the different payment statuses a payment can obtain during its life cycle please refer to the Payment Status Definitions guide.

The example below shows a possible XML reply from WorldPay for a successful authorisation of the example order, shown earlier.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4.1" merchantCode="MYMERCHANT">
    <reply>
        <orderStatus orderCode="T0211010">
             <payment>
             <paymentMethod>VISA-SSL</paymentMethod>
             <amount value="1400" currencyCode="GBP" exponent="2"
debitCreditIndicator="credit"/>
             <lastEvent>AUTHORISED</lastEvent>
             <CVCResultCode description="APPROVED"/>
                 <balance accountType="IN_PROCESS_AUTHORISED">
```

```
                      <amount value="1400" currencyCode="GBP" exponent="2"
debitCreditIndicator="credit"/>
                 </balance> <cardNumber>4444********1111</cardNumber>
                 <riskScore value="0"/>
              </payment>
         </orderStatus>
    </reply>
</paymentService>
```

In the reply, the `payment` element holds the relevant payment details and status information. Its child elements `paymentMethod` and `amount` contain the payment method used and the amount, including the currency, its exponent and the `debitCreditIndicator` that indicates the amount is positive.

The payment status is specified by the `lastEvent` element. A CVC result description is reported through the `CVCResultCode` element. The `balance` element reports on the balance in the IN_PROCESS_AUTHORISED account. For credit card payments the first and last four digits of the card number are returned in the `cardNumber` element. The `riskScore` element shows the score that the Risk Management Module assigned to the authorisation request.

Always refer to the WorldPay DTD for an up-to-date list of child elements and attributes of the `reply` element.

## Refused Response Example

A REFUSED response is given when the financial institution has refused to authorise the payment. Some of the possible reasons for refusal include: an exceeded credit limit, an incorrect expiry date, insufficient balance and exceeding the number of permitted transactions that can be made. For the full list of REFUSED Response Codes please refer to the appendix Acquirer Response Codes.

The following example is a reply where the transaction has been refused:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService
v1//EN""http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4.1" merchantCode="MYMERCHANT">
   <reply>
       <orderStatus orderCode="T0211234">
           <payment>
               <paymentMethod>ECMC-SSL</paymentMethod>
               <amount value="162095" currencyCode="GPB" exponent="2"
debitCreditIndicator="credit"/>
               <lastEvent>REFUSED</lastEvent>
               <CVCResultCode description="NOT SUPPLIED BY SHOPPER"/>
               <ISO8583ReturnCode code="33" description="CARD EXPIRED"/>
               <riskScore value="0"/>
           </payment>
       </orderStatus>
   </reply>
</paymentService>
```

Note that for a refused transaction no further processing takes place and consequently no balance information is presented. The element `ISO8583ReturnCode` shows the refusal response code from the acquirer and a mapped description (reason) from WorldPay.

*In addition to the reply message, the Payment Service can report the change of status of individual payments to your system in a number of ways. For instance, via HTTP(S) or email order notifications or via the Merchant Interface. Your system has to determine whether or not a payment was successful by interpreting this status information supplied by WorldPay.*

## PendingURL Response Example

For alternative payments supported by WorldPay AP Ltd., when a shopper is redirected to your pendingURL, you can view additional information about the transaction status.

The transaction status indicates the overall status of a transaction and shows the reason why a shopper has been redirected to your pendingURL. For example, the shopper can be redirected to a pendingURL of the following form:

```
http://www.merchant.com/pending.jsp?orderKey=ORD00XW01^MERCHANTXB^jsxml219506440&s
tatus=ERROR
```

You can use the transaction status information to manage the pending scenario appropriately, for example by allowing the shopper to retry or select another payment type if an ERROR, FAILURE, or EXPIRED status is returned.

The various transaction statuses reported by the payment method provider in the pendingURL are described in the following table.

| Status | Description |
|---|---|
| OPEN | The transaction is awaiting action by the shopper. This is the result for any offline payment method. |
| ERROR | There was a technical problem during the transaction. Some payment method providers also return this response when a shopper has cancelled their transaction. |
| FAILURE | The payment has been refused. This is an uncommon response because most alternative payment methods involve pre-funding rather than real-time authorisations. Additionally, transactions are cancelled by the shopper rather than declined by a real-time authorisation. |
| EXPIRED | The shopper session has expired. This status is returned if the shopper initiates a transaction, but does not complete it. |

For more information, see the Alternative Payment Methods Guide.

## *Informing the Shopper*

You can send an email to the shopper confirming whether the payment has been accepted or declined. Unlike an online notification, a shopper can use this information to refer to in the future. To send an email notification to the shopper you can either:

1. **Sent by merchant's system:** configure your own system to send an email in response to receiving an automated order notification from our payment service.

2. **Sent by WorldPay:** set up your WorldPay account to instruct our payment service to send an email after a successful authorisation or a refusal.

You can edit the settings and the text of the actual emails through the ‘Edit Channels’ functionality in the Merchant Interface. Please refer to our Merchant Interface User Guide for details.

# Posting an XML Order

## *Introduction*

To submit an XML order you have to set up an HTTP(S) connection to the Payment Service. How you create a connection with the WorldPay server depends on the specifications of your platform.

## *Setting-up the Connection*

*Please note that the Test and Production environments are separate environments and to connect successfully to either will require that the merchant code is correctly referenced in the XML Orders you send.*

Once you have set up the connection to the Payment Service using a valid login and password, your system has to post the XML order.

Make sure the HTTP content type is ”text/xml”! *It is important to check that 'content length' is specified correctly. Not specifying the content length will not create errors, while specifying it incorrectly will.*

The URLs to post orders to are:

- Test environment: https://secure-test.worldpay.com/jsp/merchant/xml/paymentService.jsp
- Production/Live environment: https://secure.worldpay.com/jsp/merchant/xml/paymentService.jsp

This page is intentionally blank.

# Testing the Connection

## *Testing 3-D Secure Orders*

In order for merchants to test their 3-D Secure implementation, we provide a test payment service. This service can be used to submit dummy XML orders. Please note that your WorldPay account must first be enabled for 3-D Secure, which can only be done by WorldPay support. We also provide a dummy card issuer site so that this part of the process can also be tested.

The value of the `cardHolderName` element in the XML order message can be used to manipulate the test, as follows:

| **cardHolderName value** | **Test Environment Behaviour** |
|---|---|
| 3D | The payment card is participating in 3-D Secure. The Simulator will be initiated, where further options can be selected. |
| NO3D | The payment card is not participating in 3-D Secure. The Simulator will not be initiated and the 3DS Result will be 'Authentication Offered but not Used'. |
| 3DVEERROR | The payment card is participating but simulates a system/connectivity issue that occurs before the cardholder is asked to authenticate. The Simulator page will not be initiated and the 3DS Result will be 'Authentication Unavailable'. |
| any other value | It is a normal e-commerce transaction with no 3DS initiated. |

You can use the value of the paResponse element to manipulate the outcome of the payer authentication. Using the dummy issuer site, the following options can be selected from the drop-down list:

| **paResponse value** | **Meaning** |
|---|---|
| IDENTIFIED | 'Cardholder Authenticated' |
| NOT_IDENTIFIED | 'Authentication Offered but not Used' |
| UNKNOWN_IDENTITY | 'Cardholder Failed Authentication'. The order does not proceed to authorisation. |
| CANCELLED_BY_SHOPPER | 'Cardholder Failed Authentication'. The order does not proceed to authorisation. |
| ERROR | Response failed validation checks. The order does not proceed to authorisation. |
| ERROR 3DS_VALID_ERROR_CODE | 'Authentication Unavailable'. The error code is valid. It is acceptable to proceed to authorisation. |
| ERROR 3DS_INVALID_ERROR_CODE | Response failed validation checks. The order does not proceed to authorisation. |

*Note that in the Production environment, the length of the `paResponse` element can be up to 4.7KB. However, in the Test Environment you should use considerably shorter lengths, such as those shown above, otherwise a parse error will be generated.*

## *Testing Transactions*

A number of different cases can be tested by entering the following values as the card/accountholder name (`<cardHolderName>`) in the order:

- REFUSED – will simulate a refused payment
- REFERRED - will simulate a refusal with the refusal reason 'referred'
- ERROR - will simulate a payment that ends in error.

To test your 3-D Secure implementation please refer to the section Testing 3-D Secure Orders

All other card/accountholder names will simulate an authorised payment.

For test purposes we have provided a set of test credit and debit card numbers, these are listed in the Appendices - Test Card Numbers section.

Captures and refunds can be simulated through the Merchant Interface. Use the "Capture" or "Refund" button in the Payment and Order Details page. Alternatively, you can send an XML capture or refund order modification to the Test environment.

# Submitting a 3-D Secure Order

## *Introduction*

3-D Secure is a fraud prevention measure which ensures that payments using certain credit and debit cards are authenticated by the cardholder with their bank at the time of the transaction. 3-D Secure is the common name for the technology behind MasterCard SecureCode (MSC), Verified by Visa (VbV), and American Express SafeKey (SafeKey). If your customer's card is enrolled with 3-D Secure, during the payment process they will be prompted by their bank to enter a password to verify that the transaction is authorised by the card owner before completion. For further details about authentication, refer to the Cardholder Authentication Guide.

## *Submitting a 3-D Secure Order*

This chapter describes how you should implement 3-D Secure Authentication using the XML Direct model. This process involves providing replies to two XML messages and redirecting the shopper to an authentication page. This page is provided and hosted by the shopper's Card Issuer. As this page is hosted by the shopper's card issuing bank, we have no control over its appearance or functionality.

*Please note that our Test Environment enables you to test your implementation. Please refer to the chapter Testing the Connection for further information.*

## *The Process Flow*

The figure below describes the flow of messages between the different entities involved.

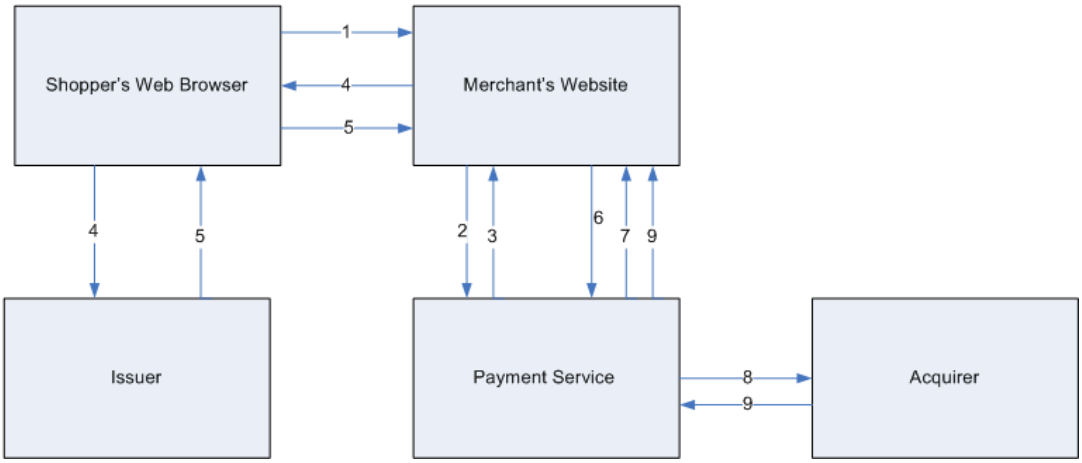


**Figure: The Process Flow**

1. The shopper places an order in the merchant's online store.
2. The merchant's system sends an XML message with the order and payment information to our system. (Refer to Initial Order Message (arrow 2).
3. WorldPay carries out a verification check to identify whether or not the cardholder is enrolled and the issuer is participating in the 3-D Secure scheme. Depending on the verification result, one of two outcomes will occur:
   - If the cardholder is enrolled in 3-D Secure a reply message is sent back to the merchant's system to request payer authentication, refer to Initial Order Reply Message (arrow 3). The process continues to step 4.
   - If the cardholder or issuer is not enrolled or the card issuer does participate in the 3-D Secure scheme then our payment service will send the order details directly to the financial institution (acquirers) for authorisation. There is no requirement for an Initial Order Reply Message (arrow 3) to be sent to complete the 3-D Secure payment process. The merchant's system will receive the normal XML response from our payment service containing the payment status of the order. Please refer to Second Order Reply Message (arrow 9).
4. The merchant's system redirects the shopper to the issuer site for authentication using information in the reply message, please refer to Redirect the Shopper to Issuer (arrow 4).
5. The authentication response is sent to the shopper, and the payer authentication response is then posted to the merchant's site.
6. The merchant includes the authentication response to the original XML order and sends it to WorldPay. It is important that your system ensures that there are no

differences between this Second Order Message (arrow 6) and the Initial Order Message (arrow 2), apart from the additional elements used to include the authentication response.

7. If the authentication response shows that the shopper failed to authenticate themselves, then one of two possible outcomes will occur as outlined below.
   - If the merchant's WorldPay account doesn't have the RMM activated, the WorldPay payment service will respond with the reply message detailed in step 4. This provides the merchant's system with the opportunity to request for the shopper to go though the payer authentication process again.
   - If the merchant's WorldPay account has the RMM activated, then the merchant's system will receive a refused response. Please refer to Second Order Reply Message (arrow 9).

*Please note that this will depend on the configuration of the Risk Management Module (RMM) of your WorldPay account. For more information on how to set up the RMM , see the Risk Management Module Guide.*

8. If the authentication response shows that the shopper was authenticated then we verify that the authentication response belongs to the authentication request and proceed with authorisation exchange with the acquirer, including the 3-D Security Authentication information.
9. After receiving the response from the acquirer, WorldPay send an authorisation response to the merchant, please refer to Second Order Reply Message (arrow 9).

## *XML Messages and HTML Redirect*

The following XML messages and HTML page are involved in the authentication process. Please Note: your system doesn't have to use an HTML form to generate the HTTP POST to the card issuer's site. This HTML form is provided as an example only of what is required to generate a submission to the card issuer's site:

- the initial order message (arrow 2 in diagram)
- reply for the initial order message (arrow 3 in diagram)
- redirect the shopper to the issuer (arrow 4 in diagram)
- the second order message with the authentication response (arrow 6 in diagram)
- reply for the second order message (arrow 8 in diagram)

## Creating an Initial Order Message (arrow 2)

The following section details the additional XML required to process a request and explains the information returned in the subsequent response.

```
<paymentService version="1.4" merchantCode="MYMERCHANT WPACC11112222">
   <submit>
       <order orderCode="T0211010" installationId="12345">
             <description>20 tulip bulbs</description>
             <amount value="2600" currencyCode= "EUR" exponent="2"/>

             <paymentDetails>
                 <VISA-SSL>
                     <cardNumber>4444333322221111</cardNumber>
                     <expiryDate>
                         <date month="09" year="2009"/>
                     </expiryDate>
                     <cardHolderName>J.Shopper</cardHolderName>
                 </VISA-SSL>
                 <session shopperIPAddress="123.123.123.123" id="021ui8ib1"/>
             </paymentDetails>
             <shopper>
                  <browser>
                      <acceptHeader>text/html,application/xhtml+xml,application/xm
l;q=0.9,*/*;q=0.8</acceptHeader>
                      <userAgentHeader>Mozilla/5.0 (Windows; U; Windows NT 5.1;
en-GB; rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5 (.NET CLR
3.5.30729)</userAgentHeader>
                  </browser>
             </shopper>
       </order>
   </submit>
</paymentService>
```

*Please note, when using the Test environment, the `cardHolderName` element must contain '3D' as the cardholder name.*

*Please note that the elements `Browser`, `acceptHeader` and `userAgentHeader` must not be hard coded by your system*

## Initial Order Reply Message (arrow 3)

This message extends the `orderStatus` element with a new sub-element; `requestInfo`. This element is a container for possible requests for information on the submitted order.

A request for 3-D Secure authentication is defined with element name `request3DSecure`. An example of the complete message is shown below.

```
<?xml version="1.0"encoding="UTF8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4"merchantCode="TECHSUPPORT">
     <reply>
         <orderStatusorderCode="WorldPay1260455114">
             <requestInfo>
                  <request3DSecure>
```

```
                        <paRequest>ThePaReq</paRequest>
                        <issuerURL><![CDATA[https://secure-
test.worldpay.com/jsp/test/shopper/VbV_TestIssuer.jsp]]></issuerURL>
                    </request3DSecure>



                </requestInfo>
            <echoData>-1374244409987691395</echoData>
            </orderStatus>
        </reply>
</paymentService>
```

The element `paRequest` contains data that was received from the 3-D Security Directory. This data must be supplied as-is in the redirect message to the shopper's issuer site.

The `issuerURL` element contains the actual URL to which the shopper must be redirected. This is shopper's Issuer site where the shopper should authenticate themselves with the 3-D Secure mechanism.

An extra element `echoData` is added, which is used by our software to process the subsequent messages belonging to the same transaction more efficiently. This element must be supplied in all subsequent messages as-is.

The session cookie that is passed back in the HTTP header of this reply message must also be extracted by your system in order that it can be returned in the HTTP header of the second order message (arrow 6), which includes the payer authentication response data. This is essential for the process to work when submitting a 3-D Secure Order.  Please refer to Capturing the Session Cookie for further information and examples.

## Redirect the Shopper to Issuer (arrow 4)

When the merchant receives the first reply with the request for 3-D Secure authentication, the merchant must redirect the shopper to the issuer's site. This must be done by submitting an HTTP POST to the Issuer URL. The HTTP POST must contain the attributes "`PaReq`" and "`TermUrl`". Optionally it can contain an attribute "`MD`".

The Issuer's site URL, to which the HTTP POST must be submitted, is given in the `issuerUrl` element of the reply message.

The value for the "`PaReq`" attribute must be the data supplied in the `paRequest` element of the reply message.

The value for the "`TermUrl`" attribute is a URL pointing to the merchant's site. It specifies the service to which the shopper is redirected from the issuer's site. The merchant is responsible for supplying a correct value.

The "MD" attribute can contain any data and will be supplied as-is in the final post when the shopper is redirected from the Issuer's site to the merchant's site (the value of the `TermUrl` attribute). This attribute can be used by the merchant to handle the session state between the original shopping session and the final post after the shopper has been authenticated.

This HTML example page redirects the shopper to the Issuer's site:

```
<html>
<head>
<title>3-D Secure helper page</title>
</head>
<body OnLoad="OnLoadEvent();">
```

```
This page should forward you to your own card issuer for identification. If your
browser does not start loading the page, press the button you see.
<br/>
After you successfully identify yourself you will be sent back to this site where
the payment process will continue as if nothing had happened.<br/>
   implemented...
<form name="theForm" method="POST" action="value of the issuerUrl element" >
<input type="hidden" name="PaReq" value="value of the paRequest element" />
<input type="hidden" name="TermUrl" value="url of merchant site" />
<input type="hidden" name="MD" value="merchant supplied data" />
<input type="submit" name="Identify yourself" />
</form>
<script language="Javascript">
<!--
     function OnLoadEvent()
{
// Make the form post as soon as it has been loaded.
document.theForm.submit();
}
// -->
</script>
</body>
</html>
```

When the shopper has enabled Javascript in the browser, the shopper will automatically forwarded to the Issuer's site. If Javascript has been disabled, the shopper must press the submit button in order to continue.

## Second Order Message (arrow 6)

The second order message is almost the same as the initial order message. Only two elements are added, the `info3DSecure` element (and sub elements) and the `echoData` element. Your system must ensure that there are no differences in the second order message other than those shown in the below example, or the second order message will be rejected by the WorldPay system.

```
<paymentService version="1.4" merchantCode="MYMERCHANT WPACC11112222">
    <submit>
        <order orderCode="T0211010
            <description>20 tulip bulbs</description>
            <amount value="2600" currencyCode= "EUR" exponent="2"/>

            <paymentDetails>
                <VISA-SSL>
                    <cardNumber>4444333322221111</cardNumber>
                    <expiryDate>
                        <date month="09" year="2009"/>
                    </expiryDate>
                    <cardHolderName>J.Shopper</cardHolderName>
                </VISA-SSL>
                <session shopperIPAddress="123.123.123.123" id="021ui8ib1"/>
                <info3DSecure>
                     <paResponse>somedata</paResponse>
                </info3DSecure>
            </paymentDetails>
            <shopper>
                <browser>
                    <acceptHeader>text/html,application/xhtml+xml,application/xml;
q=0.9,*/*;q=0.8</acceptHeader>
```

```
                    <userAgentHeader>Mozilla/5.0 (Windows; U; Windows NT 5.1; en-
GB; rv:1.9.1.5) Gecko/20091102 Firefox/3.5.5 (.NET CLR
3.5.30729)</userAgentHeader>
                </browser>
            </shopper>
             <echoData>13742444099876913 95</echoData>
        </order>
    </submit>
</paymentService>
```

The `info3DSecure` element contains the payer authentication response data received by the shopper/merchant from the issuer.

In the `echoData` element the merchant must supply the same data as it received in the first reply message.

The session cookie extracted from the initial order reply message (arrow 3) must be included in the HTTP header of the second order message. Please refer to Capturing the Session Cookie for further information and examples.

## Second Order Reply Message (arrow 9)

If a shopper fails to authenticate themselves, then you will receive a REFUSED reply. If the shopper authenticates successfully, you will receive the reply for authorised payments. Please refer to Response from the Payment Service for further information about the payment status of the order.

# *Testing 3-D Secure Orders*

In order for merchants to test their 3-D Secure implementation, we provide a test payment service. This service can be used to submit dummy XML orders. Please note that your WorldPay account must first be enabled for 3-D Secure, which can only be done by WorldPay support. We also provide a dummy card issuer site so that this part of the process can also be tested.

The value of the `cardHolderName` element in the XML order message can be used to manipulate the test, as follows:

| **cardHolderName value** | **Test Environment Behaviour** |
|---|---|
| 3D | The payment card is participating in 3-D Secure. The Simulator will be initiated, where further options can be selected. |
| NO3D | The payment card is not participating in 3-D Secure. The Simulator will not be initiated and the 3DS Result will be ‘Authentication Offered but not Used’. |
| 3DVEERROR | The payment card is participating but simulates a system/connectivity issue that occurs before the cardholder is asked to authenticate. The Simulator page will not be initiated and the 3DS Result will be ‘Authentication Unavailable’. |
| any other value | It is a normal e-commerce transaction with no 3DS initiated. |

You can use the value of the paResponse element to manipulate the outcome of the payer authentication. Using the dummy issuer site, the following options can be selected from the drop-down list.

| paResponse value | Meaning |
|---|---|
| IDENTIFIED | 'Cardholder Authenticated' |
| NOT_IDENTIFIED | 'Authentication Offered but not Used' |
| UNKNOWN_IDENTITY | 'Cardholder Failed Authentication'. The order does not proceed to authorisation. |
| CANCELLED_BY_SHOPPER | 'Cardholder Failed Authentication'. The order does not proceed to authorisation. |
| ERROR | Response failed validation checks. The order does not proceed to authorisation. |
| ERROR<br>3DS_VALID_ERROR_CODE | 'Authentication Unavailable'. The error code is valid. It is acceptable to proceed to authorisation. |
| ERROR<br>3DS_INVALID_ERROR_CODE | Response failed validation checks. The order does not proceed to authorisation. |

*Note that in the Production environment, the length of the `paResponse` element can be up to 4.7KB. However, in the Test Environment you should use considerably shorter lengths, such as those shown above, otherwise a parse error will be generated.*

## *Test Environment*

The Test environment is provided to allow merchants to test their implementation. However it is important to point out that this Test environment has some significant differences to a typical Production environment that need to be considered in implementation.

1. The issuerURL element in the Test environment contains no parameters

   http://example.issuer.url/3dsec.html

   whereas in production this URL would usually already have parameters, e.g. :

   https://example.issuer.url/pa.jsp?partner=m&CAA=B

   so that the PaReq, TermUrl and MD parameters must be posted with these parameters.

2. The acceptable values for paResponse in the Test environment (IDENTIFIED or NOT_IDENTIFIED) are significantly shorter than the values returned from the Issuer in Production where the typical length can be up to 4.7 KB. These values must be held intermediately in the merchant system for transmission to WorldPay in the second order message and require extra storage space consideration.

3. Do not submit such a long (4.7KB) paResponse to the Test environment. This will cause a parse error.

4. The redirect to the issuerURL should be always made with a POST and not a GET.

5. The PaReq value as provided must be URL-encoded before transmission to the issuer.

# Submitting a V.me by Visa Order

## *Introduction*

V.me by Visa is a secure digital wallet service provided by participating banks and supported by Visa. The digital wallet makes online shopping secure and simple and removes the need for shoppers to share card details.

From the shoppers perspective V.me by Visa saves time and provides an additional layer of security for their card details. The benefit to merchants is a higher rate of completed transactions.

For more information about V.me by visa, see: http://uk.v.me/.

This section explains how to configure your system to correctly accept V.me payments.

## *Submission of V.me Order and Payment Method Details*

The V.me XML Direct payment method is implemented differently to most XML payment methods, as you will need to redirect the shopper to V.me to allow the shopper to authenticate with V.me and agree to the payment.

The initial XML request contains the order, with the payment method details. The identifier for the V.me payment method, specified in the `paymentDetails` element, is “VME-SSL”. The V.me payment method requires three URLs to be included:

1. **successURL** this is the URL where the shopper is redirected to upon successfully completing the V.me payment. You can choose a unique URL for each transaction to find which order was paid successfully.

2. **failureURL** this is the URL where the shopper is redirected to upon unsuccessfully completing the V.me payment. You can choose a unique URL for each transaction to find which order payment was refused.

3. **cancelURL** this is the URL where the shopper is redirected to if the “cancel” or “back to merchant” link is clicked on the V.me pages. This URL can be made unique for each transaction.

***You must not include the session element (for example <session shopperIPAddress=”192.123.12.11”= id”session12345”/>) in V.me by Visa payment requests because this will result in an error response.***

The following example shows the minimum `paymentDetails` required for a V.me payment:

```
<paymentDetails>
 <VME-SSL>
  <successURL>http://worldpay.com/vme/vme_success.html</successURL>
  <failureURL>http://worldpay.com/vme/vme_failure.html</failureURL>
  <cancelURL>http://worldpay.com/vme/vme_cancel.html</cancelURL>
 </VME-SSL>
</paymentDetails>
```

*Merchants have no control over which payment methods are displayed in the V.me by Visa wallet.*

## V.me Billing Address Priority

For V.me transactions, the billing address stored with V.me (entered by the cardholder) takes precedence over any billing address submitted by you. Therefore, any billing address you supply for V.me transactions will be overridden by the address stored at V.me. The Address Verification Service (AVS) also checks the address supplied by V.me, not the address submitted by you.

## Supplying a Shopper's Email Address

You can also supply a shopper's email address. The following example shows the submission of a shopper's email address within the `shopper` element:

```
<shopper>
 <shopperEmailAddress>shopper@worldpay.com</shopperEmailAddress>
</shopper>
```

*On completion of a successful payment through the digital wallet, V.me by Visa sends its own email confirmation to the customer. In order to generate a WorldPay confirmation email to your customer, you must provide a shopper email address in the transaction details passed to the WorldPay payment page. Business Gateway merchants must supply a shopper email address.*

## Setting the Shopper Locale

This payment method requires interaction with the shopper, therefore you may wish to control the language of the V.me login screen that will be displayed to the shopper. This can be accomplished by including a `shopperLanguageCode` attribute in the order tag of the initial XML message, as shown below:

```
<order orderCode="vmetestorder123" shopperLanguageCode="en">
```

The shopper language code can be set to any valid ISO 639 language code, but only those languages supported by V.me will have an effect. Language codes not supported by V.me will cause the V.me login screen to be displayed in English.

# *V.me Responses – Redirect to V.me*

Upon receiving a request for a V.me payment, the payment service will attempt to place the order with V.me. This section describes the possible V.me responses.

## Success Response

If the payment was successful, a response similar to the following example will be returned:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="MYMERCHANT">
  <reply>
    <orderStatus orderCode="vmeexampleorder123">
      <reference id="2300036716">
        <![CDATA[https://checkout.eu.v.me/vmeservices/ServicesController/start/sta
rt?sessionId=Q0XN8PVQ6H2DAR5DPWXYP32]]>
      </reference>
    </orderStatus>
  </reply>
</paymentService>
```

The reply includes the order code, a unique numeric reference to the order, and a redirection URL for V.me. It is up to you to redirect the shopper to this URL. Note that the ampersands in URLs are escaped with SGML entities to allow them to be included in XML messages.

## Successful Response Redirect to Merchant Confirmation URL

When a successful payment is made through the V.me wallet, the shopper will be returned to your success URL, passed in your original XML request. To verify that the payment was authorised, you should use the authorised notification response. For more information, see the payment notifications guide.

## Error Response

Should there be a problem with the V.me service, the response will look like this:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="MYMERCHANT">
 <reply>
   <orderStatus orderCode="vmeexampleorder123">
     <error code="7"><![CDATA[Gateway Error: Cannot initialise V.me
payment]]></error>
   </orderStatus>
 </reply>
</paymentService>
```

This error occurs as a result of a failed connection with V.me.

This page is intentionally blank.

# Configuring your System to Receive AAV Data

## *Introduction - American Express Advanced Verification (AAV)*

American Express Advanced Verification (AAV) is a service implemented by American Express in March 2013. This service checks the details of cardholder name, telephone number and email address entered by the shopper against the equivalent records held by American Express for the Amex card used. American Express sends the result of these checks (where applicable) to WorldPay.

By default, these checks are disabled in the Risk Management Service. If you wish to enable these checks in your Risk Management Service, you will first need to:

- configure your system to receive the new values generated by the checks in your XML response.
- ensure that you capture the shopper's name, email address and telephone number on your payment page and send this information as part of your authorisation request.

*Once you have configured your system as per the instructions in this section, you can enable the checks in your Risk Management Service by emailing: corporatesupport@worldpay.com.*

## *Configuring your System to Receive AAV Data*

Depending on how your system has been configured to receive XML responses from the WorldPay payment service, you have two options for receiving AAV values:

- As a descriptor, such as: "SHOPPER DATA MATCHES"
- As a security-level single character value, such as: "A"

### Receiving AAV Data as a Descriptor

The following XML response is an example of the AAV values sent as a descriptor (highlighted in red):

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="SEPTEST1">
  <reply>
   <orderStatus orderCode="xpt-1363082509308">
    <payment>
     <paymentMethod>AMEX-SSL</paymentMethod>
     <amount value="100" currencyCode="EUR"
exponent="2"debitCreditIndicator="credit"/>
     <lastEvent>AUTHORISED</lastEvent>
     <CVCResultCode description="APPROVED"/>
     <AVSResultCode description="APPROVED"/>
     <AAVAddressResultCode description="SHOPPER DATA MATCHES"/>
```

```
     <AAVPostcodeResultCode description="SHOPPER DATA MATCHES"/>
     <AAVCardholderNameResultCode description="SHOPPER DATA MATCHES"/>
     <AAVTelephoneResultCode description="DATA NOT SENT"/>
     <AAVEmailResultCode description="SHOPPER DATA MATCHES"/>
      <cardHolderName>
       <![CDATA[asd]]>
      </cardHolderName>
     <issuerCountryCode>GB</issuerCountryCode>
     <balance accountType="IN_PROCESS_AUTHORISED">
      <amount value="100" currencyCode="EUR" exponent="2"
debitCreditIndicator="credit"/>
     </balance>
     <cardNumber>3742*******0001</cardNumber>
     <riskScore value="21"/>
    </payment>
    <date dayOfMonth="14" month="03" year="2013" hour="10" minute="22"
second="1"/>
   </orderStatus>
  </reply>
</paymentService>
```

## Range of AAV Descriptor Values Returned by WorldPay

This table lists the possible AAV descriptor values returned by WorldPay:

| Value | Description of Value |
|---|---|
| SHOPPER DATA MATCHES | The data entered by the shopper matches the data held by American Express for the card used. |
| SHOPPER DATA DOES NOT MATCH | The data entered by the shopper does not match the data held by American Express for the card used. |
| DATA NOT SENT | American Express has not received either the telephone number, email address or cardholder name. This may not have been entered by the shopper. |
| DATA NOT CHECKED BY ACQUIRER | American Express has not checked either the telephone number, email address or cardholder name. |
| UNKNOWN | Due to unknown circumstances (for example, an error), a check was not able to be made. |

*Ensure that you capture the shopper's name, email address and telephone number on your payment page and send this information as part of your authorisation request.*
*Once you have configured your system to receive AAV check data, you need to email: corporatesupport@worldpay.com to enable the checks in your Risk Management Service.*

## Receiving AAV Data as a Single-Character Value

The following XML response example highlights the AAV values sent as a single-character value (highlighted in red):

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.bibit.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="SEPTEST1">
  <reply>
   <orderStatus orderCode="xpt-1363082509308">
    <payment>
     <paymentMethod>AMEX-SSL</paymentMethod>
     <amount value="100" currencyCode="EUR" exponent="2"
debitCreditIndicator="credit"/>
     <lastEvent>AUTHORISED</lastEvent>
     <CVCResultCode description="APPROVED"/>
     <AVSResultCode description="APPROVED"/>
     <AAVAddressResultCode description="A"/>
     <AAVPostcodeResultCode description="A"/>
     <AAVCardholderNameResultCode description="A"/>
     <AAVTelephoneResultCode description="C"/>
     <AAVEmailResultCodedescription="A"/>
     <cardHolderName>
      <![CDATA[asd]]>
     </cardHolderName>
     <issuerCountryCode>GB</issuerCountryCode>
     <balance accountType="IN_PROCESS_AUTHORISED">
      <amount value="100" currencyCode="EUR" exponent="2"
debitCreditIndicator="credit"/>
     </balance>
     <cardNumber>3742*******0001</cardNumber>
     <riskScore value="21"/>
    </payment>
    <date dayOfMonth="14" month="03" year="2013" hour="10" minute="25"
second="1"/>
   </orderStatus>
  </reply>
</paymentService>
```

### Range of AAV Values Returned by WorldPay

This table lists the possible AAV single-character values returned by WorldPay:

| Value | Description of value |
|---|---|
| A | Data matched |
| B | Data not checked |
| C | Data not supplied |
| D | Data not matched |

*Ensure that you capture the shopper's name, email address and telephone number on your payment page and send this information as part of your authorisation request.*
*Once you have configured your system to receive AAV check data, you need to*

*email: corporatesupport@worldpay.com to enable the checks in your Risk Management Service.*

# Appendices

## *Payment Method Codes*

The merchant can use the `paymentMethodMask` or the `preferredPaymentMethod` variable to determine which payment methods the shopper can use. Codes for the payment methods can be found in the tables below.

For the full list of payment methods, see the Document Type Definition (DTD), available at: Document Type Definition (DTD) for XML Integration.

For information about alternative payment methods that are supported by WorldPay AP Ltd., see the Alternative Payment Methods Guide.

### Credit Cards

| Name | Payment Method Code | Area | Remarks |
|---|---|---|---|
| American Express SSL | AMEX-SSL | International | N/A |
| VISA | VISA-SSL | International | Visa Credit/Debit/Electron |
| MasterCard | ECMC-SSL | International | The name **Eurocard** is no longer in use. |
| AirPlus | AIRPLUS-SSL | International | N/A |
| Aurore | AURORE-SSL | International | N/A |
| Carte Bancaire | CB-SSL | France | N/A |
| Carte Bleue | CARTEBLEUE-SSL | France | N/A |
| Dankort | DANKORT-SSL | Denmark | N/A |
| Diners | DINERS-SSL | International | N/A |
| Discover Card | DISCOVER-SSL | United States | N/A |
| GE Capital | GECAPITAL-SSL | International | N/A |
| Japanese Credit Bank | JCB-SSL | International | N/A |
| Laser Card | LASER-SSL | Ireland | N/A |
| PayPal | PAYPAL-EXPRESS | International | Card/Wallet |
| UATP | UATP-SSL | International | N/A |

## Online Debit Methods

| Name | Payment Method Code | Area | Remarks |
|---|---|---|---|
| Elektronisches Lastschriftverfahren | ELV-SSL | Germany | N/A |
| Maestro | MAESTRO-SSL | UK | Depending upon the issuer policy, either the issuer number or the start date may need to be included in the paymentDetails. |

## Offline Payment Methods

| Name | Payment Method Code | Area | Remarks |
|---|---|---|---|
| Direct Bank Transfer<br><br>Redirect Bank Transfer | TRANSFER_AT-BANK | Austria | Bank Transfer |
| | TRANSFER_BE-BANK | Belgium | |
| | TRANSFER_DK-BANK | Denmark | |
| | TRANSFER_FI-BANK | Finland | |
| | TRANSFER_FR-BANK | France | |
| | TRANSFER_DE-BANK | Germany | |
| | TRANSFER_GR-BANK | Greece | |
| | TRANSFER_IT-BANK | Italy | |
| | TRANSFER_JP-BANK | Japan | |
| | TRANSFER_LU-BANK | Luxembourg | |
| | TRANSFER_NL-BANK | Netherlands | |
| | TRANSFER_NO-BANK | Norway | |
| | TRANSFER_PL-BANK | Poland | |
| | TRANSFER_ES-BANK | Spain | |
| | TRANSFER_SE-BANK | Sweden | |
| | TRANSFER_CH-BANK | Switzerland | |
| | TRANSFER_GB-BANK | United Kingdom | |
| Giropay | GIROPAY-SSL | Germany | N/A |

| Name | Payment Method Code | Area | Remarks |
|---|---|---|---|
| Signed Direct Debit | PERMANENT_SIGNED_DD | Germany, Netherlands, Spain, and USA | N/A |
| Unsigned Direct Debit | SINGLE_UNSIGNED_DD | Germany, Netherlands, Spain, and USA | N/A |

### Online Alternative Payment Methods

| Name | Payment Method Code | Area | Remarks |
|---|---|---|---|
| China Union Pay | CHINAUNIONPAY-SSL | International | N/A |
| ENETS | ENETS-SSL | Singapore | Bank Transfer |
| Swedbank | HANSABANK-SSL | Sweden | Bank Transfer |
| IDEAL | IDEAL-SSL | Dutch | Bank Transfer |
| PayPal | PAYPAL-EXPRESS | International | E-wallet |

### Online Digital Wallets

| Name | Payment Method Code | Area | Remarks |
|---|---|---|---|
| V.me | VME-SSL | Europe | Digital wallet service |

## *ISO Currency Codes*

Currencies accepted by the WorldPay Payment Service are listed below.

Please note that the values in the orders sent to WorldPay do not have any decimal delimiters. Merchants should use 'exponent' instead. Exponent is the number of decimals available in the currency. Also note that currency code is always in capitals.

In the following example the amount payable by the shopper is Euro 19,82:

```
<amount value="1982" currencyCode="EUR" exponent="2"/>
```

The full ISO 4217 list can be found at: www.iso.org. WorldPay does not take responsibility for an external link's operation or content.

| Code | Name | Exponent |
|---|---|---|
| ARS | Nuevo Argentine Peso | 2 |

| | | |
|---|---|---|
| AUD | Australian Dollar | 2 |
| BRL | Brazilian Real | 2 |
| CAD | Canadian Dollar | 2 |
| CHF | Swiss Franc | 2 |
| CLP | Chilean Peso | 2 |
| CNY | Yuan Renminbi | 2 |
| COP | Colombian Peso | 2 |
| CZK | Czech Koruna | 2 |
| DKK | Danish Krone | 2 |
| EUR | Euro | 2 |
| GBP | Pound Sterling | 2 |
| HKD | Hong Kong Dollar | 2 |
| HUF | Hungarian Forint | 2 |
| IDR | Indonesian Rupiah | 0 |
| ISK | Iceland Krona | 2 |
| JPY | Japanese Yen | 2 |
| KES | Kenyan Shilling | 2 |
| KRW | South-Korean Won | 0 |
| MXP | Mexican Peso | 2 |
| MYR | Malaysian Ringgit | 2 |
| NOK | Norwegian Krone | 2 |
| NZD | New Zealand Dollar | 2 |
| PHP | Philippine Peso | 2 |
| PLN | New Polish Zloty | 2 |
| PTE | Portuguese Escudo | 2 |
| SEK | Swedish Krone | 2 |
| SGD | Singapore Dollar | 2 |
| SKK | Slovak Koruna | 2 |
| THB | Thai Baht | 2 |
| TWD | New Taiwan Dollar | 2 |
| USD | US Dollar | 2 |
| VND | Vietnamese New Dong | 2 |
| ZAR | South African Rand | 2 |

# ISO Country Codes

The `countryCode` element is used in XML orders/communications, it is an upper-case two-letter 'ISO 3166' standard country code, as shown in the following example:

```
  ...
<address>
    <countryCode>GB</countryCode>
</address>
```

The full list can be found at: www.iso.org. WorldPay does not take responsibility for an external link's operation or content.

# Acquirer Response Codes

WorldPay uses the ISO 8583 Response Codes in the orderStatusEvent messages to indicate the status of the payment: AUTHORISED, REFUSED, etc. The Payment Service maps these responses to a simplified list. Below you will find all possible response codes, their numeric value and the mapping to a status. For further information about the different payment statuses a payment can obtain during its life cycle please refer to the Payment Status Definitions guide.

## ISO 8583 Response Codes

| Card Message Value | Status | Code Message Value | Status |
|---|---|---|---|
| 0 AUTHORISED | AUTHORISED | 85 REJECTED BY CARD ISSUER | REFUSED |
| 2 REFERRED | REFUSED | 91 CREDITCARD ISSUER TEMPORARILY NOT REACHABLE | REFUSED |
| 4 HOLD CARD | REFUSED | 97 SECURITY BREACH | REFUSED |
| 5 REFUSED | REFUSED | 3 INVALID ACCEPTOR | ERROR |
| 8 APPROVE AFTER IDENTIFICATION | REFUSED | 12 INVALID TRANSACTION | ERROR |
| 13 INVALID AMOUNT | REFUSED | 14 INVALID ACCOUNT | ERROR |
| 15 INVALID CARD ISSUER | REFUSED | 19 REPEAT OF LAST TRANSACTION | ERROR |
| 17 ANNULATION BY CLIENT | REFUSED | 20 ACQUIRER ERROR | ERROR |

| | | | |
|---|---|---|---|
| 28 ACCESS DENIED | REFUSED | 21 REVERSAL NOT PROCESSED, MISSING AUTHORISATION | ERROR |
| 29 IMPOSSIBLE REFERENCE NUMBER | REFUSED | 24 UPDATE OF FILE IMPOSSIBLE | ERROR |
| 33 CARD EXPIRED | REFUSED | 25 REFERENCE NUMBER CANNOT BE FOUND | ERROR |
| 34 FRAUD SUSPICION | REFUSED | 26 DUPLICATE REFERENCE NUMBER | ERROR |
| 38 SECURITY CODE EXPIRED | REFUSED | 27 ERROR IN REFERENCE NUMBER FIELD | ERROR |
| 41 LOST CARD | REFUSED | 30 FORMAT ERROR | ERROR |
| 43 STOLEN CARD, PICK UP | REFUSED | 31 UNKNOWN ACQUIRER ACCOUNT CODE | ERROR |
| 51 LIMIT EXCEEDED | REFUSED | 40 REQUESTED FUNCTION NOT SUPPORTED | ERROR |
| 55 INVALID SECURITY CODE | REFUSED | 58 TRANSACTION NOT PERMITTED | ERROR |
| 56 UNKNOWN CARD | REFUSED | 64 AMOUNT HIGHER THAN PREVIOUS TRANSACTION AMOUNT | ERROR |
| 57 ILLEGAL TRANSACTION | REFUSED | 68 TRANSACTION TIMED OUT | ERROR |
| 62 RESTRICTED CARD | REFUSED | 80 AMOUNT NO LONGER AVAILABLE, AUTHORISATION EXPIRED | ERROR |
| 63 SECURITY RULES VIOLATED | REFUSED | 92 CREDITCARD TYPE NOT PROCESSED BY ACQUIRER | ERROR |
| 75 SECURITY CODE INVALID | REFUSED | 94 DUPLICATE REQUEST ERROR | ERROR |

| | | | |
|---|---|---|---|
| 76 CARD BLOCKED | REFUSED | | |

# *Security Code and Address Verification Checks and Responses*

You can carry out Security Code (CVC/CVV) and Address Verification (AVS) checks on an individual direct order.

These fraud prevention tools provide a mechanism for checking the authenticity of a transaction by comparing information entered by the shopper during the payment process, with details held by the card issuer. Only XML orders containing a valid  code fragment will be checked by our payment service.

The example below shows an example of a CVC coded fragment.

```
    <cardHolderName>J. Shopper</cardHolderName>
    <cvc>123</cvc>
    <cardAddress>
```

## Testing

The following CVC/CVV scenarios can be tested using the codes listed below.

1. To test CVC and AVS:

| **CVC/CVV Code** | **Simulated Situation** |
|---|---|
| Left blank | NOT SUPPLIED BY SHOPPER |
| 111 | NOT SENT TO ACQUIRER |
| 222 | NO RESPONSE FROM ACQUIRER |
| 333 | NOT CHECKED BY ACQUIRER |
| 444 | FAILED |
| 555 | APPROVED |

2. To test CVC with AMEX:

| CVC/CVV Code | Simulated Situation |
|---|---|
| Left blank | NOT SUPPLIED BY SHOPPER |
| 1111 | NOT SENT TO ACQUIRER |
| 2222 | NO RESPONSE FROM ACQUIRER |
| 3333 | NOT CHECKED BY ACQUIRER |
| 4444 | FAILED |
| 5555 | UNKNOWN |
| 6666 | APPROVED |

3. To test AVS (using the billing postcode address):

| AVS Code | Simulated Situation |
|---|---|
| Left blank | NOT SUPPLIED BY SHOPPER |
| 1111 | NOT SENT TO ACQUIRER |
| 2222 | NO RESPONSE FROM ACQUIRER |
| 3333 | NOT CHECKED BY ACQUIRER |
| 4444 | FAILED |
| 5555 | UNKNOWN |
| 6666 | APPROVED |

## *Test Card Numbers*

You can use the following card numbers to test transactions in the test environment only. When using test cards, you can specify an expiry date up to seven years in the future. The test cards do not have a card verification code and issue number.

| Card Type | Card Number |
|---|---|
| Airplus | 122000000000003 |
| American Express | 34343434343434 |

| | |
|---|---|
| Cartebleue | 5555555555554444 |
| Dankort | 5019717010103742 |
| Diners | 36700102000000 and 36148900647913 |
| Discover card | 6011000400000000 |
| JCB | 3528000700000000 |
| Laser | 630495060000000000 and 630490017740292441 |
| Maestro (UK only) | 6759649826438453 |
| Mastercard | 5555555555554444 and 5454545454545454 |
| Visa | 4444333322221111, 4911830000000, and 4917610000000000 |
| Visa Delta - Non-UK | 4462030000000000 |
| Visa Electron (UK only) | 4917300800000000 |
| Visa Purchasing | 4484070000000000 |

Visa Purchasing transactions are treated as Visa credit card transactions.

## *German ELV*

To test German ELV payments in the test environment a correctly formatted account number (Kontonummer) and valid bank code (Bankleitzahl) should be used, for example:

Account number: 12345678
Bank code: 10000000
Bank name: Bundesbank
Bank residence: Berlin

| Payment Method | Bank Code | Account Number |
|---|---|---|
| ELV | 20030000 | 92441196 |
| ELV | 43050001 | 122108525 |
| ELV | 30070024 | 5929120 |

If you want to test ELV transactions, ensure that ELV is activated in your production environment.

# *XML Error Codes*

The list of XML error codes is as follows:

- Error Code 1 - Internal error, a general error
- Error Code 2 - Parse error, invalid XML
- Invalid number of transactions in batch
- Error Code 4 - Security error
- Error Code 5 - Invalid request
- Invalid content, occurs when XML is valid but content of XML is not
- Error Code 7 - Payment details in the order element are incorrect
- V.me Error Code 7 - Connection to V.me failed

For full details of the WorldPay Document Type Definition (DTD), please refer to:

http://dtd.worldpay.com/paymentService_v1.dtd

## Examples

The following are some examples for these error codes.

### Error Code 1 - Internal error, a general error

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.WorldPay.om/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="MYMERCHANT WPACC11112222">
<reply>
<error code="1"><![CDATA[IInternal error]]></error>
</reply>
</paymentService>
```

It is difficult to define internal errors as they may be caused by a number of things. Internal errors have their origin within the WorldPay system itself. When a merchant gets this error it is best to retry after a short while. When a serious internal error occurs WorldPay's technical staff are informed automatically and the problem will be corrected.

### Error Code 2 - Parse error, invalid XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.WorldPay.om/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="MYMERCHANT WPACC11112222">
<reply>
<error code="2"><![CDATA[Empty body in message]]></error>
</reply>
</paymentService>
```

This error indicates that the body of the XML message posted was empty. This error is also returned when the 'content length' has been set incorrectly, i.e. too few characters have been specified.

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.WorldPay.om/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="MYMERCHANT WPACC11112222">
<reply>
<error code="2"><![CDATA[Missing DOCTYPE declaration]]></error>
</reply>
</paymentService>
```

This error indicates that the XML code sent to WorldPay does not contain the required doctype declaration. This is used by our payment service to determine what kind of information is being sent.

## Error Code 4 - Security error

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="MYMERCHANT WPACC11112222">
<reply>
<error code="4"><![CDATA[Security Violation.]]></error>
</reply>
</paymentService>
```

This error code usually indicates one of the following: (1) there is a difference between the merchant code used to set up the connection and that referred to in the XML message, (2) a connection has been attempted from an unregistered IP, or (3) the merchant is submitting to an inactive environment, usually because they have only activated the Test environment, but are attempting to submit to production.

## Error Code 5 - Invalid request

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService
v1//EN""http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="MYMERCHANT WPACC11112222">
<reply>
<orderStatus orderCode="123456">
<error code="5"><![CDATA[Duplicate Order]]></error>
</orderStatus>
</reply>
</paymentService>
```

Each orderCode has to be unique. In this example the merchant tried to post an order with the orderCode 123456 to our payment service. However, this order for the merchant already exists in the WorldPay database. A simple way to make orderCodes unique is to use a date/time-stamp, an incremental number or a combination of both.

## Error Code 7 - Payment details in the order element are incorrect

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="MYMERCHANT WPACC11112222">
<reply>
```

```
<orderStatus orderCode="1112">
<error code="7"><![CDATA[Invalid payment details : Expiry date =
01/2002]]></error>
</orderStatus>
</reply>
</paymentService>
```

The example shows a payment that has been refused, because the expiry date occurs in the past.

## V.me Error Code 7 - Connection to V.me failed

The following XML example shows an error code 7 specific to V.me:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService v1//EN"
"http://dtd.worldpay.com/paymentService_v1.dtd">
<paymentService version="1.4" merchantCode="DEMO">
<reply>
<orderStatus orderCode="TEST123">
<error code="7"><![CDATA[Gateway Error: Cannot initialise V.me payment]]></error>
</orderStatus>
</reply>
</paymentService>
```

This error occurs as a result of a failed connection with V.me.