Business Gateway

XML Direct Integration Guide

V6.5 June 2019

Use this guide to:

- Integrate with Worldpay
- Create and test XML Direct orders
- Implement and test 3D Secure
- Look up ISO codes, payment method codes, and more

Contents

1 Introduction

The **Business Gateway: XML Direct Integration Guide** describes how to integrate your payment platform with Worldpay's payment gateway using the XML Direct model.

This guide shows you:

- How to create, validate and submit an XML order
- How to test your integration with Worldpay
- What responses you can expect to receive from Worldpay's payment gateway
- How to implement the 3D Secure fraud prevention in the XML Direct model

To help you create, test and manage your XML orders, this guide also provides you with a range of reference materials, including test card numbers. For more information, seethe appendices at the end of this guide.

1.1 What is XML Direct?

The XML Direct model enables online merchants who collect their shoppers' payment details and selected payment method on their own platform to process payments through Worldpay.

A direct integration means that:

- Your shoppers make their payment on your website, instead of being redirected to the Worldpay payment pages
- You keep full control of the payment process, including the payment pages that are displayed to shoppers

1.2 Is your business ready?

The technical complexity and costs involved in implementing an XML Direct integration (including PCI DSS compliance) means that the XML Direct model is only suitable for those merchants with established high transaction volumes.

Before you can integrate with Worldpay using the XML Direct model you must demonstrate that:

- Your systems can collect and store payment data securely
- You are taking responsibility for your PCI DSS compliance

Worldpay will require evidenceof:

- A clean Vulnerability scan of your systems
- A successful assessment for PCI DSS compliance

For more information about PCI DSS compliance, see **2.2.1 Payment Card Industry Data Security Standard (PCI DSS)**

For an overview of how the XML Direct model works, see **2 Overview***.*

1.3 Who is this guide for?

This is a technical integration guide, aimed at:

- System integrators
- Othertechnical roles, including managers, who are involved in designing and managing your payments solution

1.3.1 Skills and knowledge

To carry out the tasks described in this guide, you require the following skills and knowledge:

- XML programming skills
- Knowledgeof HTTPS
- Basic knowledgeof the Worldpay payment services

For more information about Worldpay's payment service, including payment methods, see the Worldpay website at **http://www.worldpay.com***.*

1.4 More help?

For more information about Worldpay's products and services, including payment methods, seethe Worldpay website at **http://www.worldpay.com**

For technical documentation, see **http://www.worldpay.com/support/bg/**

Developerresources (including the Worldpay DTD) are located on the Corporate Gateway > Guides and Resources pages.

For more information, see **http://www.worldpay.com/support/gg/**

To contact Worldpay support:

- Email: **support@worldpay.com**
- Phone: +44 (0)870 3661233

1.5 Legal

This document and its content are proprietary to Worldpay and may not be reproduced, published or resold. The information is provided on an “AS IS” basis for information purposes only and Worldpay makes no warranties of any kind including in relation to the content or suitability. Terms and Conditions apply to all our services.

Worldpay (UK) Limited (Company No: 07316500/ FCA No: 530923), Worldpay Limited (Company No: 03424752 / FCA No: 504504), Worldpay AP Limited (Company No: 5593466 / FCA No: 502597). Registered Office: The Walbrook Building, 25 Walbrook, London EC4N 8AF and authorised by the Financial Conduct Authority under the Payment Service Regulations 2009 for the provision of payment services. Worldpay (UK) Limited is authorised and regulated by the Financial Conduct Authority for consumer credit activities.

Worldpay, the logo and any associated brand names are all trademarks of the Worldpay group of companies.

2 Overview

This chapter provides an overview of the XML Direct model of integration with Worldpay's payment service.

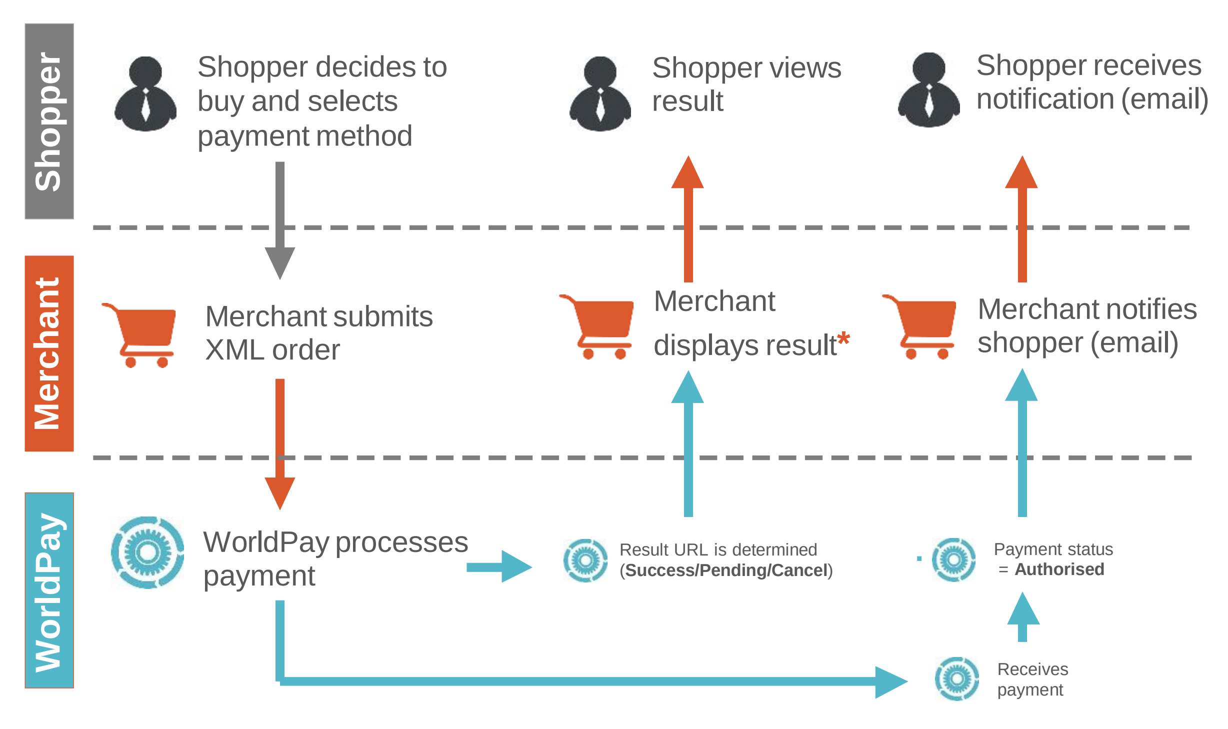
It describes:

- Why you might want to choose the XML Direct model to integrate with Worldpay
- How payments are processed through Worldpay (payment flow), using the XML Direct model
- How the security of payments is protected, through the PCI DSS security initiative and 3D Secure authentication

2.1 Why XML Direct?

The XML Direct model is an XML-based method of integrating your website with Worldpay's payment service. You may choose this model if:

- You want to collect more of your shoppers data
- You want to manage of the of the shopper journey, within your own environment


Shopper
Merchant
WorldPay
Shopper decides to buy and selects payment method
Shopper views result
Shopper receives notification (email)
Merchant submits XML order
Merchant displays result*
Merchant notifies shopper (email)
WorldPay processes payment
Result URL is determined (Success/Pending/Cancel)
Payment status = Authorised
Receives payment

*To optimise the shopper experience, WorldPay recommends providing your own website result URLs.

Figure 1: XML Direct

XML (Extensible Markup Language) is a universal way of exchanging data across applications and platforms. Worldpay uses XML to send encrypted messages about payments between your system and our payment service over the Internet. To learn more, go to **http://www.w3.org/XML/**.

2.1.1 What the merchant does

In the XML Direct model, the merchant:

- Collects details of the items that the shopper wants to purchase
- Collects the shopper's mandatory payment details (including cardholder names and card numbers) and their selected payment method
- Takes payment through their website. The shopper is not redirected to Worldpay's payment pages to submit their payment. The merchant can also specify the website URLs the shopper must go to after they complete their payment

2.1.2 What Worldpay does

In the XML Direct model, Worldpay:

- Processes the shopper's payment order
- Carries out any change requests from the merchant (for example, to refund or cancel the order).
- Responds to queries about the status of the order from the merchant (for example, to find out if the order has been Authorised, Captured or Settled)

2.2 Securing payments in the XML Direct model

To collect and store payment information, such as card numbers and cardholder names, your systems must fully comply with the Payment Card Industry Data Security Standard (PCI DSS). You may also want to reduce your exposure to fraud and increase confidence in online shopping by implementing 3D Secure authentication.

To integrate with Worldpay using the XML Direct model you must demonstrate that:

- Your systems can collect and store payment data securely
- You are taking responsibility for your PCI DSS compliance

Worldpay will require evidence of:

- A clean **Vulnerability scan** of your systems
- A successful assessment for **PCI DSS compliance**

The costs involved in implementing the appropriate security measures for PCI DSS compliance means that the XML Direct model is only suitable for those merchants with established high transaction volumes.

2.2.1 Payment Card Industry Data Security Standard (PCI DSS)

The Payment Card Industry Data Security Standard (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 way. PCI DSS:

- Combines the security standards for cardholder data at Mastercard and Visa
- Is endorsed by American Express, JCB and Diners Club

A major focus for PCI DSS is the technology that is used to collect, store and process card data. This makes PCI DSS compliance particularly important for merchants operating the XML Direct model, who collect and store payment details on their own systems.

For more information about PCI DSS, including its hardware and software standards, seethe PCI Security Standards website at **http://www.pcisecuritystandards.org**.

*To help you comply with PCI DSS, the PCI Security Standards website also lists PCI-approved Quality Security Assessors (QSAs), who can advise on your system's security (a chargeable service). Worldpay is **not** responsible for the content or operation of external websites.*

2.2.2 3D Secure authentication

3D Secure is a mandatory authentication scheme for online credit and debit card transactions. The scheme is designed to:

- Help reduce your exposure to fraud
- Increase confidencein online shopping through an additional level of authentication

The benefits of implementing 3D Secure include ashift in liability in the event of fraudulent transactions.

The additional security benefits and liability shifts of authenticated transactions are currently only supported by Visa, Mastercard, and American Express SafeKey.

To learn more about 3D Secure orders, see **7 Submitting a 3D Secure order**.

3D Secure implementations are branded to the relevant card scheme and card issuer:

Card scheme	3D Secure implementation
Visa	Verified by Visa
Mastercard	Mastercard SecureCode
American Express (UK and Singapore only)	American Express SafeKey

Table 1: Card scheme implementations of 3D Secure

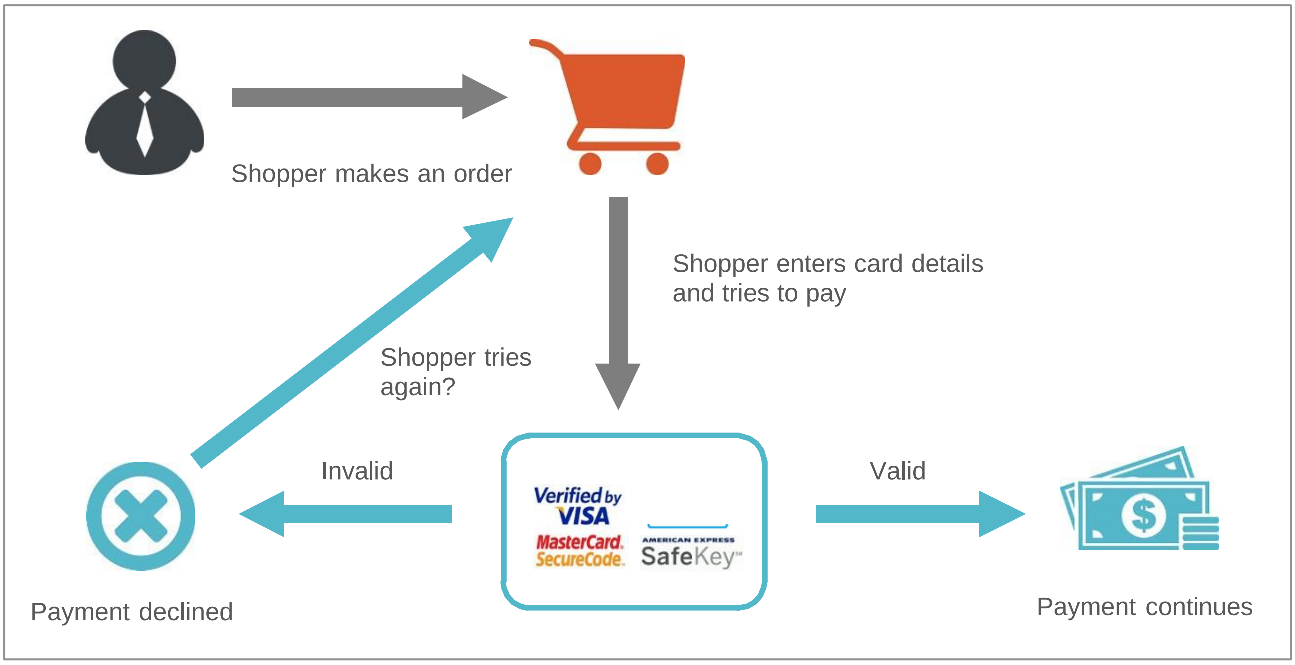


Figure 2: 3D Secure authentication

3D Secure is currently limited to Internet payments, and does not cover:

- Fax, mail, or phone orders
- All card types

2.2.3 MCC 6012 Merchants and VISA

From **1 June 2014**, you must send us extra information for domestic payments processed in the United Kingdom if you are under MCC (Merchant Category Code) 6012.

MCC 6012 covers a range of payments for financial services. Examples of this type of payment include paying off all or part of a balance on a credit card or loan, or repayment of a mortgage.

This change applies even if you have additional merchant codes as well as MCC 6012.

Merchants assigned the code MCC 6012 must collect the following information for each UK domestic VISA transaction. The information is the primary recipient's:

- Account Number / Primary Account Number (PAN)
- Last name (family name)
- Date of Birth (D.O.B)
- Postcode

Primary recipients are the entities (people or organisations) that have a direct relationship with the financial institution. Also, these primary recipients have agreed to the terms and conditions of the financial institution.

For more details of this requirement, see **Section 4.5**.

Failure to comply may cause VISA to fine you.

3 Integrating with Worldpay

This chapter outlines the major tasks you must carry out to integrate your website with Worldpay's payment service, using the XML Direct model. These tasks include:

- Setting up a connection between your website and Worldpay, using HTTPS
- Creating the valid XML files that are used to communicate with Worldpay
- Testing your integration

3.1 Connecting using HTTPS

HTTPS adds the security capabilities of the Secure Sockets Layer (SSL) encryption protocol to standard HTTP communications.

To submit XML messages safely and securelyto Worldpay's payment service, you must set up a connection between your website and Worldpay using HTTPS.

To set up your connection using HTTPS:

1. You must register your domain with an SSL certificates provider.
2. Worldpay sent you your XML username and password when you opened your account. If you can't find them, contact Worldpay support to get them resent.
3. Use your XML login and password to submit XML messages.

   To change your Worldpay XML password, contact **support@worldpay.com**.

4. Create valid XML messages that you can use to submit orders, order modifications and status inquiries to Worldpay (see **3.2 Creating and submitting valid XML messages**).
5. Set up your platform for submitting XML messages to Worldpay's payment service.

   Example scripts for ASP, Java, Java Servlet and PHP based platforms are available from the Worldpay website > Support Centre at
   http://www.worldpay.com/support/gg/index.php?page=examples&c=WW

6. Submit your XML messages:
   - To the test environment at **https://secure-test.worldpay.com/jsp/merchant/xml/paymentService.jsp**
   - To the production (live) environment at **https://secure.worldpay.com/jsp/merchant/xml/paymentService.jsp**

Before you can submit XML messages, the test and production environments must be activated by Worldpay. You should also check that:

- The HTTPS content type is "text/xml"
- The content length is specified correctly. Not specifying the content length willnot create errors, but specifying it incorrectly will

The Worldpay payment service only accepts incoming XML messages if the originating IP address is registered for the merchant. The IP address to connect to the production environment can only be changed by Worldpay.

3.1.1 Error Code 4 – Security error

When they try to connect to the Worldpay payment service forthe first time, merchants sometimes experience an **Error Code 4 – Security Error**.

This error code usually indicates one of the following issues:

Issue type	Issue
XML login	The automatically generated password (XMLlogin) that was used to set up the connection and the automatically generated password referenced by the XML message do not match.
XML password	The XML password set up by the merchant in the Merchant Interface and the XML password provided by the XMLmessage do not match.
IP address	The originating IP address forthe XML message is not registered forthe merchant.
Environment	The merchant is submitting XMLmessages to an inactive environment. This is usually because the merchant has only activated the test environment, but is trying to submit messages to the production environment.

Table 2: Error Code 4 – Security error

For the full list of error codes, see **Appendix G: XML error codes**.

3.2 Creating and submitting valid XML messages

The XML orders you submit to the Worldpay payment service must:

- Use correct XML syntax and conform to the Worldpay Document Type Definition (DTD)
- Contain content that complies with your contract with Worldpay, and is not more than 4k in size

3.2.1 Worldpay DTD

The Worldpay DTD provides all the XML elements that you require for communicating with the Worldpay payment service and third party processors. It includes detailed comments on the use of elements, and the structure of valid XML messages.

The DTD includes elements (not a definitive list) for:

- Payment orders and order modifications (for example, messages to cancel or refund an order)
- 3D Secure orders and order modifications
- FuturePay payments (repeat payments, used for subscriptions and other regular payments)
- Payment status inquiries (for example, to check if an order has been Authorised, Captured or Settled)
- Communicating with alternative payment methods (the non-card based payment methods supported by Worldpay)

The Worldpay DTD is available to view and reference from **http://dtd.worldpay.com/v1/***.*

You can also download the DTD from the ***Worldpay website > Support Centre*** *at* **http://www.worldpay.com/support/gg/index.php?page=guides&c=WW***.*

3.2.2 Valid XML

All the XML messages you sent to the Worldpay payment service must be valid.

Well-formedness

Your XML is well-formed if:

- Every start tag [<exampletag>] has a matching end tag [</exampletag>]
- Elements do not overlap
- There is only one root element [<paymentService>]
- Attribute values are always presented within quotes [exampleattribute value=”23”]
- Elements do not have two attributes with the same name
- Comments and processing instructions do not appear inside tags
- No unescaped [<] or [&] signs occur in the element or attribute’s character data

Reference the DTD

A valid XML message always includes a reference to the DTD (in this case, the Worldpay DTD [paymentService_v1.dtd]), so that the message can be checked against the DTD automatically.

For more information about the referencing the Worldpay DTD in your messages, see **4.1 XML and DTD declarations***.*

Specify the Installation ID

Worldpay recommends including the installation ID in your XML messages.

You must include the installation ID (for example, installationId=”12345”) within the submit element when submitting orders using the XML Direct model. You can find the installation ID in the **Merchant Interface > Profile > Installations > Installation ID**.

For more information about including the installation ID in your orders, see **4.3.2 installationId attribute***.*

Use declared elements only

Every element, attribute and entity in the XML that you send to the Worldpay payment service must be declared in the DTD (in this case, the Worldpay DTD).

XML elements can be declared to contain the following:

XML data types	Description
NMTOKEN	Name tokens
PCDATA	Parsed character data
CDATA	Character data or constants

Table 3: XML element declared

Name tokens (NMTOKEN)

An XML name token [NMTOKEN] consists of:

- Alphanumeric and/or ideographic characters
- The punctuation marks [_], [-], and [:]

No other characters are allowed. An XML name token cannot contain spaces. If an attribute is declared in the DTD to contain name tokens, the values of that attribute must be valid XML name tokens. For example:

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

Code example 1: valid XML name tokens

PCDATA

You cannot include the following special characters in a PCDATA (Parsed Character DATA) section in the XML message: [&], [<], [>] and [“].

If you want to use these characters, then you must use the equivalent hexadecimal character code:

Character	Hexidecimal character code
&	&amp;
>	&gt;
<	&lt;
“	&quot;

Table 4: Hexadecimal character codes

CDATA

You can include any data / characters in a CDATA (Character DATA) section, provided that the data / characters:

- Comply with the specified encoding
- Do not contain the following character set (which is used to express the end tag):]]>

You must enclose a CDATA section between the start tag and the end tag. For example:

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

Code example 2: CDATA section

4 Structure of an XML Direct order

This chapter describes how to create a valid XML Direct order.

All XML messages sent to Worldpay's payment servicemust:

- Reference the Worldpay DTD at **http://dtd.worldpay.com/v1/**
- Always use the correct XML syntax and conform to the DTD (be valid XML)

The content of XML orders must comply with your contract with Worldpay, and not exceed 4KB in size.

For general guidance on the Worldpay DTD and creating valid XML messages, see **3.2 Creating and submitting valid XML messages***.*

4.1 XML and DTD declarations

All valid, well-formed XML files used in the XML Direct integration model begin with an XML declaration.

They must also contain a document type declaration, containing the root element `paymentService` and reference to the public Worldpay payment service DTD (paymentService_v1.dtd).

The paymentService root element must also include the version number of the Worldpay DTD (in this case, v1).

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE paymentService PUBLIC "-//WorldPay//DTD WorldPay PaymentService
v1//EN"

"http://dtd.worldpay.com/paymentService_v1.dtd">
```

Code example 3: XML and DTD declaration

4.2 Merchant code

You have the option to specify a merchant code within the `paymentService` root element. If you:

- Specify a merchant code (for example, merchantCode="WPACC11112222") then Worldpay will process the payment using that code. The merchant code is always spelt out in capitals and must be the same as the one you used as your login name (see **3.1 Connecting using HTTPS**)
- Do not specify a merchant code, then Worldpay automatically selects the first available merchant code that is relevant to the payment

The merchant code that is selected depends on card and account configuration details, such as a specific currency or payment method. For more information, contact **support@worldpay.com***.*

4.2.1 Limitations on merchant code matching

The merchant code matching facility cannot be applied according to a chosen capture delay specification.

If a merchant code is not specified as part of the `paymentService` root element then a payment is automatically placed in the first relevant merchant code that has a capture delay of zero. Merchant codes with a capture delay set as **off** are selected last.

To place an order within a merchant code with a specific capture delay setting, you must specify a particular merchant code within the `paymentService` root element.

*To find your merchant code(s), go to **Merchant Interface > Profile > Identification**.*

```
<paymentService  version="1.0"  merchantCode="  WPACC11112222">
      <submit>
          [Order  information  goes  here]
      </submit>
</paymentService>
```

Code example 4: DTD version and Merchant Code

4.3 The order element

The `order` element is:

- Found within the `submit` element
- Used to describe the goods or services that the shopper is ordering

If your shopper's browser doesn't have cookies enabled, it is possible that another person can access the shopper's session ID. This is because the URL contains the session ID. We recommend that you ask your shoppers to enable cookies on their web browsers. Cookies minimise the opportunity to record a session ID and misuse it.

4.3.1 orderCode attribute

The `orderCode` attribute is a required attribute of the `order` element. The `orderCode` attribute:

- Must have a unique value
- Can be up to 64 characters in length. Spaces, quotation marks, code brackets (< and >) and pipes ("|") are not allowed

An order with a previously used order code cannot be processed correctly. If you use a previously used order code, you will receive error messages and have problems with reporting.

You can use the `orderCode` attribute to contain a Cart ID if the Cart ID is unique. If the Cart ID is not unique, then you must use the `orderCode` attribute with a unique number added to the static Cart ID.

4.3.2 installationId attribute

The installationId attribute:

- Must be included within the submit element when submitting orders using the XML Direct model
- Is an attribute of the order element

You can find the installation ID in the **Merchant Interface > Profile > Installations > Installation ID**.

4.3.3 description and amount

The first two order child elements are description and amount:

Child element	Description
description	The description element is used to contain a simple, one-line description of the order (up to 255 characters long).
amount	The amount element has three attributes: • value, which specifies the total amount the shopper is expected to pay • currencyCode, which specifies the currency (ISO 4217 code) • exponent, which specifies where the decimal point or comma should be placed in the value, counting from the right

Table 5: description and amount child elements

For a list of currency codes and their respective exponents, see **Appendix B: ISO currency codes**.

```
<order  orderCode=”T0211010”  installationId="12345">
     <description>20  red  roses  from  the  MyMerchant  webshop.</description>
     <amount  currencyCode=”GBP”  exponent=”2”  value=”5000”/>
</order>
```

Code example 5: Order element and child elements

Example note:

The content is highlighted in red.

4.3.4 orderContent child element

The third order child element is `orderContent`. The `orderContent` child element is used to contain the order content. You can deliverthe content of the order in HTML format. When supplying HTML order content:

- You must place all HTML tags between the `<body>` and `</body>` tags of a valid HTML document
- You cannot use scripting in the order content

Always place the order content in a CDATA section to avoid parsing problems:

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

Code example 6: orderContent

Example note:

The content is highlighted in red.

4.3.5 paymentDetails child element

The `paymentDetails` child element is the fifth `order` child element. The `paymentDetails` element contains the details of the selected payment method.

To enable the Worldpay payment service to submit a 3D Secure transaction successfully, the `paymentDetails` element must also include information about the shopper's browser session.

The child element session (which contains the `shopperIPAddress` and `session ID` elements) contains the shopper's browser session information.

Every payment method has its own set of elements and attributes. For the list of available payment method codes for the XML Direct model, including child elements, see:

- *The Worldpay DTD at* **http://dtd.worldpay.com/v1/**
- **Appendix A: Payment method codes**

Worldpay uses the payment details and session information for risk assessment. This information is also a mandatory element in 3D Secure orders.

paymentDetails example 1: Visa payment

The following example shows a Visa payment, where VISA-SSL is the payment method code:

```
<paymentDetails>
   <VISA-SSL>
      <cardNumber>4444433333322221111</cardNumber>
            <expiryDate> <date month=”09” year=”2009”/> </expiryDate>
               <cardHolderName>J. Shopper</cardHolderName>
                  <cvc>123</cvc>
                  <cardAddress>
                        <address>
                              <address1>47A</address1>
                              <address2>Queensbridge Road</address2>
                              <address3>Suburbia</address3>
                              <postalCode>CB94BQ</postalCode>
                               <city>Cambridge</city>
                              <state>Cambridgeshire</state>
                              <countryCode>GB</countryCode>
                              <telephoneNumber>0122312345</telephoneNumber>
                         </address>
              </cardAddress>
     </VISA-SSL>
<session  shopperIPAddress=”123.123.123.123”  id=”0215ui8ib1”  />
</paymentDetails>
```

Code example 7: paymentDetails: Visa payment

Example notes:

The content is highlighted in red.

`<VISA-SSL>` The payment method code VISA-SSLis used for both Visa credit and Visa debit card payments.

`<CVC>` The CVC element contains the Card Verification Code.

paymentDetails example 2: MasterPass payment

The following example shows a MasterPass payment, where MASTERPASS-SSL is the payment method code:

```
<paymentDetails>
    <MASTERPASS-SSL>

<successURL>http://worldpay.com/masterpass/masterpass_success.html</successURL>

<failureURL>http://worldpay.com/masterpass/masterpass_failure.html</failureURL>

<cancelURL>http://worldpay.com/masterpass/masterpass_cancel.html</cancelURL>
    </MASTERPASS-SSL>
</paymentDetails>
```

Code example 8: paymentDetails: MasterPass payment

The MasterPass payment method is implemented differently to most XML payment methods. For more information see **8 Submitting a MasterPass order**.

Example notes:

The content is highlighted in red.

<MASTERPASS-SSL> The MASTERPASS service uses the payment method code MASTERPASS-SSL

<URL> The URL that relates to a success, failure or cancel outcome.

4.3.6 shopper child element

The `shopper` child element is the sixth `order` child element. The shopper element contains further information about the cardholder making the payment.

It includes the `shopperEmailAddress` element, which is used by the Worldpay payment service to:

- Identify possible fraudulent transactions
- Send an email to the shopper when the payment is authorised or refused

To redirect the shopper to the correct card issuer site for 3D Secure authentication, the `shopper` element must also include information about the shopper's browser settings (using the elements browser, `acceptHeader` and `userAgentHeader`).

The shopper is only redirected for 3D Secure authentication if Worldpay can confirm that the shopper has enrolled with the 3D Secure scheme.

shopper example: Firefox browser information

```
<shopper>
      <shopperEmailAddress>jshopper@myprovider.int</shopperEmailAddress>
         <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>
```

Code example 9: shopper: Firefox browser information

Example notes:

The content is highlighted in red.

<acceptHeader> The `acceptHeader` element must contain exactly the same content as the HTTP accept header that was sent to the merchant by the shopper's user agent.

<userAgentHeader> The `userAgentHeader` element must contain exactly the same content as the HTTP user-agent header that was sent to the merchant by the shopper's user agent.

4.3.7 statementNarrative

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

```
<statementNarrative>Statement narrative text goes here</statementNarrative>
```

Code example 10: statementNarrative

Example note:

The content is highlighted in red.

Support for the statementNarrative element is currently restricted to a limited number of payment methods and acquirers. For more information, contact **support@worldpay.com**.

4.4 Example XML Direct order

The following example shows a complete order for the XML Direct model:

```
<?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="1.0" merchantCode="WPACC11112222">
     <submit>
         <order orderCode=”T0211010” installationId=”12345”>
           <description>20 red roses from the MyMerchant webshop.</description>
              <amount currencyCode=”GBP” exponent=”2” value=”5000”/>
                <paymentDetails>
                      <VISA-SSL>
                         <cardNumber>44444433333322221111</cardNumber>
                         <expiryDate> <date month=”09” year=”2009”/></expiryDate>
                         <cardHolderName>J. Shopper</cardHolderName>
                         <cvc>123</cvc>
                          <cardAddress>
                              <address>
                                 <address1>47A</address1>
                                 <address2>Queensbridge Road</address2>
                                 <address3>Suburbia</address3>
                                 <postalCode>CB94BQ</postalCode>
                                 <city>Cambridge</city>
                                 <state>Cambridgeshire</state>
                                 <countryCode>GB</countryCode>
                                 <telephoneNumber>0122312345</telephoneNumber>
                               </address>
                          </cardAddress>
                         </VISA-SSL>
                   <session shopperIPAddress=”123.123.123.123” id=”0215ui8ib1” />
                 </paymentDetails>
            <shopper>
```

```
               <shopperEmailAddress>jshopper@myprovider.int</shopperEmailAddress>
   <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>
<statementNarrative>Statement narrative text goes here</statementNarrative>
            </order>
      </submit>
</paymentService>
```

Code example 11: Example XML order

Example note:

The content is highlighted in red.

Business Gateway merchants must submit an email address for the shopper to receive a Worldpay email confirmation.

Before you submit XML messages to the Worldpay payment service, we strongly recommend that you validate the XML your system creates.

*XML that does not conform to the Worldpay DTD (***http://dtd.worldpay.com/v1/***) is not accepted. For more information about creating valid XML messages, see* **3.2 Creating and submitting valid XML messages***.*

There are a number of tools you can use to check and validate XML. For example, see **http://xml.coverpages.org/check-xml.html**

4.5 Important Information for MCC 6012 Merchants

You must make this change if you have the merchant code 6012 (Financial institution – manual and automated, securities broker or dealer, insurance sales, insurance premiums, insurance carrier) and process UK domestic payments.

This change applies even if you have additional merchant codes assigned to you, as well as MCC 6012.

4.5.1 Information to Collect

Merchants with an MCC 6012 code must collect the following information for each transaction

The Account Number/ Primary Account Number (PAN) of the Primary Recipient

The PAN (a unique identifier) must belong to the primary recipient. The primary recipient is the person or entity who has the direct relationship with the financial institution and has agreed to its terms and conditions. The primary recipient may or may not be the person or entity that makes the payment.

When you collect the Account Number/PAN, you must format the field in the following way:

Card to card payments (for example, use a card to pay off a card) – Send the first six digits and the last four digits of the recipient's PAN with no spaces. For example FFFFFFLLLL

Card to non-card payments (for example, pay off a loan) – Send the first 10 characters of the recipient account number.

This field must only contain letters or digits, or can be left empty. If the information is not available, leave this field empty.

Don't use special characters in the Account Number field.

Last Name

Use letters only, do not use digits. Use standard English characters – avoid punctuation marks like accents and circumflexes.

Date of Birth (DOB)

Use the format DD-MM-YYYY (day, month, year)

Some merchants collect the month and year of birth only. If you do this, please modify your system so that you collect the day of birth along with the month and year.

If you cannot provide the day of birth, please modify your system so it cannot pass the date of birth value to Worldpay as part of the MCC 6012 changes. Do not pass an empty value in the tag, or not send the date of birth field at all. This is because an incomplete date of birth value may cause an increase in rejected transactions.

The date of birth must always be in the past – use digits only.

Postcode

A valid UK postcode in the format:

- AA9A 9AA
- A9A 9AA
- A9 9AA
- A99 9AA
- AA9 9AA
- AA99 9AA

Postcodes must have a space between the first and last group of characters/numbers. Send the above details to the Worldpay WPG system. See section **4.5.2** below.

Once we receive these details, we send them to the card issuer to screen as part of the transaction process.

Implement these changes as soon as possible; merchants who don't may get a fine.

4.5.2 MCC 6012 Technical Information

DTD (Document Type Definition) changes

The paymentService_v1 DTD contains elements specific to MCC 6012;this means you can pass the primary recipient's data to us at Worldpay.

If you integrate with Worldpay using CG Redirect, and you are using MCC6012, you must always send us the additional data for primary recipients. This is because when the order is created, payment method is unknown.

The new elements are highlighted in red:

```
<!ELEMENT order   (  description,
                                  amount,
                                  risk?,
                                  orderContent?,
                                  (paymentMethodMask | paymentDetails | payAsOrder ),
                                  shopper?,
                                  shippingAddress?,
                                  billingAddress?,
                                  branchSpecificExtension?,
                                  redirectPageAttribute?,
                                  paymentMethodAttribute*,
                                  echoData?,
                                  statementNarrative?,
                                  hcgAdditionalData?,
                                  thirdPartyData?,
                                  shopperAdditionalData?) >


<!-- Used to collect merchant-held data
     required by Visa for MCC6012 merchants -->


<!ELEMENT shopperAdditionalData (shopperAccountNumber?, lastName?, birthDate?,
postalCode?)>
<!ELEMENT shopperAccountNumber (#PCDATA)>
```

Code example 12: MCC 6012 additional fields

The `lastName`, `birthDate` and `postalCode` elements already exist in the DTD.

All the new tags are optional – Worldpay does not monitor whether an MCC6012 merchant passes the data or not. You must ensure that you capture the data and send it to Worldpay.

Format of the MCC 6012-specific fields

Field	Description
shopperAccountNumber	This field can be empty. It contains a maximum of 10 characters. It must contain only letters or digits. This shopper account number represents the unique account identifier of the primary recipient. This can be a partial PAN (Primary Account Number) number: so you must send the first 6 digits + last 4 digits, for example FFFFFFLLLL. You can also send up to 10 characters from the account number. If the content of this field does not follow these rules, you receive this error message: *The shopperAccountNumber cannot be longer than 10 characters or The shopperAccountNumber must contain only digits or letters.*
lastName	This field can be empty. It must not contain digits. If the content of the lastName does not follow the rules above, you receive this error message: *Digits are not allowed in lastName tag.*
birthDate	This must be a valid date in the past. It is best to supply the day, month and yearin the format DD-MM-YYYY. If you are unable to supply the day value in the date of birth, then do **not** send the D.O.B (date of birth). A blank or partially supplied date of birth may cause an increase in declined transactions. If the content of the `birthDate` is not correct, you receive an error message.
postalCode	This field can be empty. It must be a valid UK postal code (one of the format: AA9A 9AA, A9A 9AA, A9 9AA, A99 9AA, AA99AA, AA99 9AA). A valid postcode always has a blank space between the two groups of letters and numbers. If the content of the postal code is not correct, you receive this error message: *The postalCode must contain a valid UK postalcode.*

Table 6: Format of MCC 6012 specific fields

Example of the correct XML code

```
<?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">
 <submit>
  <order orderCode="jsxml390799671">
   <description>&amp;nbsp;</description>
   <amount value="100" currencyCode="EUR" exponent="2"/>
   <orderContent>
   </orderContent>
   <paymentMethodMask>
    <include code="ALL"/>
   </paymentMethodMask>
   <shopper>
    <shopperEmailAddress>sp@worldpay.com</shopperEmailAddress>
   </shopper>
   <shippingAddress>
    <address>
     <firstName>John</firstName>
     <lastName>Doe</lastName>
     <street>The Science Park</street>
     <houseNumber>270</houseNumber>
     <postalCode>CB4 0WE</postalCode>
     <city>Cambridge</city>
     <countryCode>GB</countryCode>
    </address>
   </shippingAddress>
    <shopperAdditionalData>
    <shopperAccountNumber>1234ABC</shopperAccountNumber>
    <lastName>Oana</lastName>
    <birthDate>
      <date dayOfMonth="10" month="10" year="2000"/>
    </birthDate>
    <postalCode>CB4 0WE</postalCode>
   </shopperAdditionalData>
  </order>
 </submit>
</paymentService>
```

Code example 13: An example of the additional fields for MCC 6012 merchants

5 Responses to an XML Direct order

This chapter describes the XML responses that are sent to you by the Worldpay payment system when you submit an XML order.

When the Worldpay payment service receives a valid order with payment details, the payment service sends that information to the financial institutions (acquirers) for authorisation. The result of the authorisation request is reported to Worldpay as either:

- **AUTHORISED**
- **REFUSED**

If there is a problem with the order, an **ERROR** response is sent.

In the XML Direct model, Worldpay then sends an XML response back to your system about the payment status of the order.

To parse XML responses from the Worldpay payment service, you must use an industry standard XML parser.

Homemade parsers may not be able to correctly interpret the messages Worldpay sends you. For more information about XML parsers, see **http://www.xml.org**.

For more information about the different payment statuses that a payment can be given during its life cycle, see the ***Payment Status Definitions Guide****.*

For a list of payment status response codes, see **Appendix D: Acquirer response codes***.*

Warning and Caution alerts

Risk Management results (Warning and Caution alerts) are **not** shown in the response from Worldpay to an XML Direct order. Risk Management results are available from either:

- The **Merchant Interface > Payment Details** page
- The **Payment Notifications (callback) service**. You can set up and modify this service in the **Merchant Interface > Installations** page

5.2 Example AUTHORISED reply message

An AUTHORISED reply message is sent by Worldpay when the financial institution (acquirer) has approved the payment. The following example shows a reply from Worldpay after a payment has been successfully authorised:

```
<?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="WPACC11112222">
```

```
        <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>
```

Code example 14: AUTHORISED reply from Worldpay

Example note:

The content is highlighted in red.

*Athough an AUTHORISED response is a strong indication that the payment details that were submitted are valid, it is **not** a guarantee of payment. For more information, see the **Payment Status Definitions Guide**.*

5.2.1 Key to example AUTHORISED reply

Child element	Description
payment	The `payment` element contains the relevant payment details and status information for the order.
amount	The `amount` element contains the: ● `value`, which specifies the total amount the shopper is expected to pay ● `currencyCode`, which specifies the currency (ISO 4217 code) ● `exponent`, which specifies where the decimal point or comma should be placed in the value, counting from the right ● `debitCreditIndicator`, which indicates that the amount is positive ("credit")
LastEvent	The `LastEvent` element specifies the payment status (AUTHORISED).
CVCResultCode	The `CVCResultCode` element reports the result of the **Card Verification Code (CVC)** check ("APPROVED").

Child element	Description
balance	The balance element reports on the balance in the account (“IN_PROCESS_AUTHORISED”).
cardNumber	For credit card payments, the first and last four digits of the card number are returned in the cardNumber element.
riskScore	The riskScore element shows the score that the Risk Management Module assigned to the authorisation request (“0”).

Table 7: Key to example AUTHORISED reply from Worldpay

For the full list of the reply element’s child elements and attributes, see the Worldpay DTD at **http://dtd.worldpay.com/v1/**.

5.3 Example REFUSED reply message

A REFUSED reply is sent by Worldpay when the financial institution (the acquirer) has refused to authorise the payment.

Reasons for refusing a payment include the shopper having gone over their credit limit, an incorrect expiry date, and insufficient funds. Fora full list of REFUSED response codes, see **Appendix D: Acquirer response codes**.

The following example shows a reply from Worldpay after a payment 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="WPACC11112222">
       <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>
```

Code example 15: Example REFUSED reply

Example note:

The content is highlighted in red.

Because no more processing takes place after a payment has been refused, a REFUSED reply message does not present any balance information. In the above example, the element `ISO8583ReturnCode` shows:

- The refusal response code from the acquirer ("33")
- The mapped description (reason) from Worldpay ("CARD EXPIRED")

5.4 Other ways of reporting changes to payments

As well as the reply message, the Worldpay payment service can report the status of individual payments to your system using:

- HTTPS
- Email order notifications
- The Merchant Interface

Your system has to determine if a payment was successful by interpreting the status information supplied by Worldpay.

5.5 Payment statuses in the pendingURL

You can view additional information about the transaction status where the shopper:

- Has used an alternative payment method supported by Worldpay AP Ltd
- Has been redirected to your pendingURL

The transaction status shows you:

- The overall status of the payment
- The reason why the shopper was 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&status=ERROR
```

Code example 16: pendingURL

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.

For more information about alternative payment methods, see the ***Alternative Payment Methods Guide****.*

5.5.1 Transaction statuses in the pendingURL

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 responsewhen 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 • Transactions are usually 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.

Table 8: pendingURL transaction statuses

5.6 Telling the shopper about the status of a payment

A merchant can send an email to the shopper confirming whether the payment has been accepted or declined.

Unlike an online notification, a shopper can keep this information for their records. To send an email notification to the shopper you can either:

- Send the email from your system. To do this, you must configure your own system to send an email in response to an automated order notification from our payment service
- Send the email from Worldpay. To do this, you must set up your Worldpay account so that it instructs our payment service to send an email after a successful authorisation or a refusal

If you would like Worldpay to send the email notification, email **support@worldpay.com**.

*When this feature has been activated, you can edit the settings and the text of the email notifications by going to the **Merchant Interface > Edit Channels**.*

6 Submitting a batch order

Instead of sending Worldpay orders for processing individually, you can submit a large number of orders in one batch.

You can:

- Submit batch orders to Worldpay at any time of day
- Submit a batch of individual orders or recurring payments. Each batch order should ideally contain between 100 and 3000 individual orders
- Send a batch modification to cancel the order batch
- Perform a batch inquiry to find out the status of the batch and the payment status

For more information about batch inquiries and modifications, see the ***Order Modifications and Inquiries Guide****.*

Sending batch orders is right for you if:

- You do not need immediateonline feedback on the status of orders
- The selected payment method requires little orno interaction with the shopper, after the order has been placed (for example, an offlinepayment with Giropay, ora debit payment with Solo)
- Your business model allows you to store large numbers of orders securely on your own platform, and send them to Worldpay at regular intervals for processing

Your systems must be secure to collect and store payment details in compliance with Payment Card Industry Data Security Standards (PCI DSS). For more information see **2.2.1 Payment Card Industry Data Security Standard (PCI DSS).**

6.1 orderBatch element

In batch orders, the `submit` element contains an `orderBatch` element. The `orderBatch` element contains multiple `order` elements, which in turn contain information about the goods or services that have been ordered.

The `orderBatch` elements has two attributes:

Attribute	Values
transactions	The number of individual orders in the batch.
merchantBatchCode	A batch identifier, which must be unique.

Table 9: orderBatch attributes

```
<orderBatch  transactions=”300”  merchantBatchCode=”B0123”>
</orderBatch>
```

Code example 17: orderBatch attributes

Example notes:

The content is highlighted in red.

`transactions` The number of orders in this batch order is 300.

`merchantBatchCode` The unique identifier for this batch order is B0123.

For more information about the `order` element and its child elements (including `description` and `amount`), see **4.3 The order element**.

6.2 Example batch order

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay/DTD WorldPay
PaymentServicev1//EN"
"http://dtd.worldpay.com/paymentService_v1.dtd">
      <paymentService version="1.4" merchantCode="MYMERCHANT">
        <submit>
           <orderBatch transactions="3" merchantBatchCode="B1234">
              <order orderCode="T0011011">
              <description>20 tulip bulbs from MYMERCHANT Webshop</description>
              <amount value="2600" currencyCode="EUR" exponent="2"/>
              <orderContent>
                <![CDATA[order content here]]>
              </orderContent>
              <paymentDetails>
                 <VISA-SSL>
                   <cardNumber>4444333322221111</cardNumber>
                   <expiryDate><date month="09" year="2007"/></expiryDate>
                   <cardHolderName>J.Shopper</cardHolderName>
                   <cvc>123</cvc>
                   <cardAddress>
                      <address>
                         <firstName>John</firstName>
                         <lastName>Shopper</lastName>
                          <address1>11 Shopperstreet</address1>
                          <address2>Shopper suburb</address2>
                          <address3>Shoppervillage</address3>
                          <city>Shoppercity</city>
                          <region>Shoppercounty</region>
                          <postalCode>1234</postalCode>
                          <countryCode>NL</countryCode>
                          <telephoneNumber>0123456789</telephoneNumber>
                      </address>
                    </cardAddress>
                  </VISA-SSL>
                 <session shopperIPAddress="123.123.123.123" id="0215ui8ib1"/>
               </paymentDetails>
             <shopper>
             <shopperEmailAddress>jshopper@myprovider.int</shopperEmailAddress>
             </shopper>
            </order>
          <order orderCode="T0011012">
            <description>A model windmill from MYMERCHANT Webshop</description>
            <amount value="14300" currencyCode="EUR" exponent="2"/>
            <orderContent>
              <![CDATA[order content here]]>
            </orderContent>
            <paymentDetails>
                 <SINGLE_UNSIGNED_DD_NL-SSL>
                 <BankAccount_NL>
                      <bankAccountNr>1234568</bankAccountNr>
                    <accountHolderName>Jan Klant</accountHolderName>
                    <accountHolderResidence>Amsterdam</accountHolderResidence>
                <BankAccount_NL>
                </SINGLE_UNSIGNED_DD_NL-SSL>
              <session shopperIPAddress="111.112.113.114" id="7613tu8iq9"/>
            </paymentDetails>
         </order>
           <order orderCode="T0011014">
             <description>3 pairs of wooden shoes from MYMERCHANT Webshop</description>
             <amount value="9800" currencyCode="EUR" exponent="2"/>
             <orderContent>
                <![CDATA[order content here]]>
             </orderContent>
             <paymentDetails>
               <ELV-SSL>
                 <accountHolderName>Johannes Käufer</accountHolderName>
                 <bankAccountNr>1234567</bankAccountNr>
                 <bankName>My bank</bankName>
                 <bankLocation>Berlin</bankLocation>
                 <bankLocationId>12345678</bankLocationId>
               </ELV-SSL>
              <session shopperIPAddress="456.456.456.456" id="7613tu8iq9"/>
             </paymentDetails>
           <shopper>
           <shopperEmailAddress>j.kaufer@meinprovider.deu</shopperEmailAddress>
           </shopper>
         </order>
       </orderBatch>
     </submit>
</paymentService>
```

Code example 18: batch order

Example notes:

The content is highlighted in red.

`transactions`

The number of orders in this example is 3. Ideally, the number of orders contained in a batch order is between 100 and 3000.

merchantBatchCode
The unique identifier for this batch order is B1234.

<orderContent>
To make it easier to read, the order content has been left out of this example.

<VISA-SSL>
<SINGLE_UNSIGNED_DD_NL-SSL>
<ELV-SSL>
The orders contained in a batch order can be paid for using a variety of different payment methods (in this case, a Visa card, an unsigned Dutch direct debit, and ELV respectively).

6.3 Example response to a batch order

```
<?xml version="1.0"?>
<!DOCTYPE paymentService PUBLIC "-//WorldPay/DTD WorldPay PaymentService v1//EN"
"http://dtd.worldpay.com/paymentService_v1.dtd">
     <paymentService merchantCode="MYMERCHANT" version="1.4.1">
       <reply>
         <batchStatus transactions="3" merchantBatchCode="B1234"
status="ORDERS_SAVED"/>
       </reply>
     </paymentService>
```

Code example 19: response to a batch order (batchStatus)

Example notes:

The content is highlighted in red.

When Worldpay receives avalid and correctly formatted batch order, a reply message is sent that confirms that the status of the batch (batchStatus) is ORDERS_SAVED. The batch order will be processed at a scheduled time.

The orders from the batch are processed individually and invalid orders will generate individual error messages. The batch order is given another batch status when the payments have been processed. For more information, see **Table 10: Batch order statuses** below.

6.3.1 Batch order statuses

Attribute	Values
ORDERS_SAVED	This status indicates that Worldpay: ● Has saved the batch for processing at a scheduled time ● Was able to parse the XML in the batch order The orders from the batch are processed individually. Invalid orders will generate individual error messages.

Attribute	Values
CANCELLED	You can cancel batch orders with the status ORDERS_SAVED by sending an XML batch modification (seethe **Order Modifications and Inquiries Guide**). The orders in a cancelled batch will not have been processed and will have no payment status.
PROCESSED	This status indicates that all orders within the batch have been processed and that no errors were encountered
PROCESSED_WITH_ERRORS	This status indicates that the orders within the batch have been processed but some errors were encountered.

Table 10: Batch order statuses

For more information about batch inquiries and modifications, see the ***Order Modifications and Inquiries Guide****.*

7 Submitting a 3D Secure order

This chapter describes how to implement 3D Secure, a mandatory authentication scheme for credit and debit card transactions, in the XML Direct model.

To submit a 3D Secure order:

- You must provide replies to two XML messages
- Redirect the shopper to an authentication page, provided and hosted by the shopper's card issuer

For a brief overview of 3D Secure authentication, including supporting card schemes, see **2.2.2 3D Secure authentication**.

Because the 3D Secure authentication page is hosted by the card issuing bank, Worldpay has no controlover the appearance and functionality of the page.

7.1 How does 3D Secure work in the XML Direct model?

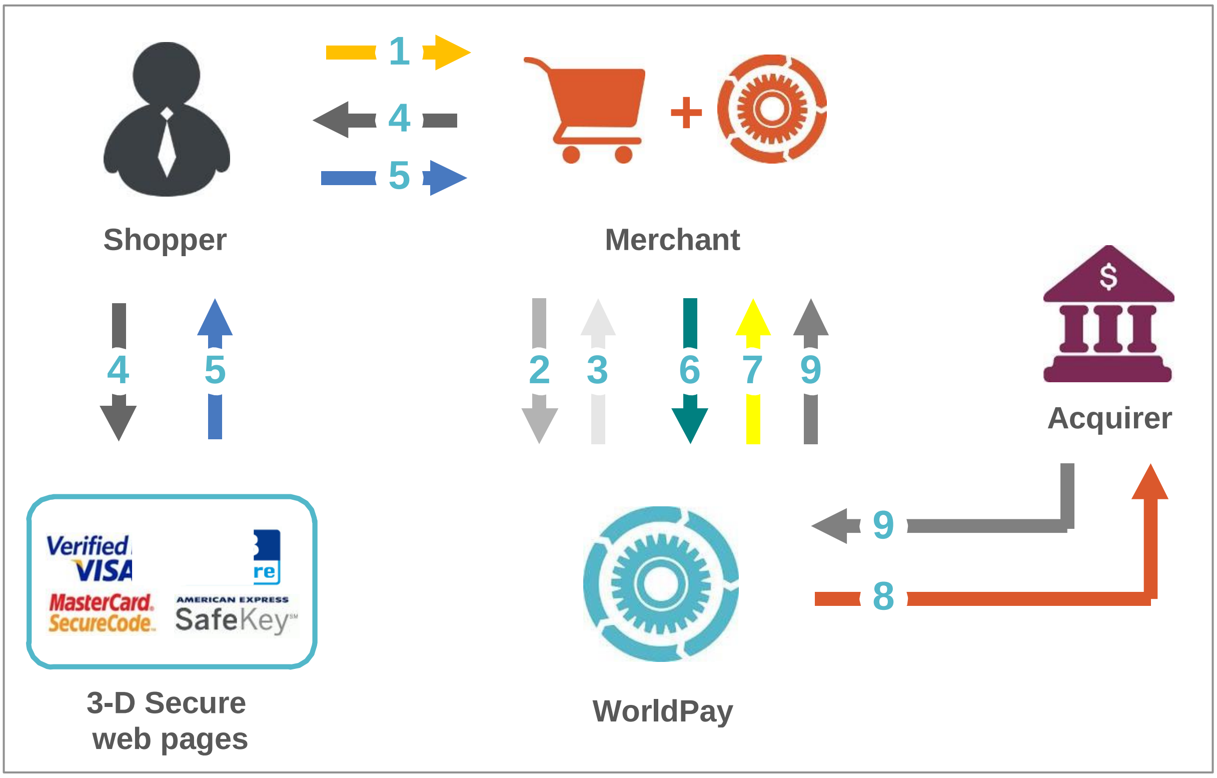


Figure 3: 3D Secure process flow

7.1.1 Key to Figure 4: 3D Secure process flow

Step / arrow	Description
▬ 1 ➡	The shopper places an order in the merchant's online store.
▬ 2 ➡	The merchant's system sends an initial XML message with the order and payment information to the Worldpay payment service.
▬ 3 ➡	Worldpay carries out a verification check to identify if: • The cardholder is enrolled in the 3D Secure scheme • The card issuer is participating in the 3D Secure scheme **Outcome 1:** If the card issuer is participating in the 3D Secure scheme, and the cardholder is enrolled in the 3D Secure scheme, a message is sent to the merchant’s system to request payer authentication. The process continues to step / arrow 4. **Outcome 2:** If the card issuer is not participating in the 3D Secure scheme, or the cardholder is not enrolled, the Worldpay payment service sends the order details directly to the acquirer for authorisation. The merchant’s system is sent the normal XML response by the Worldpay payment service, containing the payment status of the order. See step / arrow 9.
▬ 4 ➡	The merchant's system redirects the shopper to the issuer site for 3D Secure authentication, using information in the reply message.
▬ 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 adds the authentication response to the original XML order and sends it to Worldpay. **Note:** There should be no differences between the first XML order message (step/arrow 2) and the second XML order message (step/arrow 6), except for the additional elements used to contain the authentication response.

Step / arrow	Description
7 ➡	If the authentication response shows that the shopper failed to authenticate themselves, then the merchant's system receives a REFUSED response. **Note:** If the merchant's Worldpay account has the Risk Management Module (RMM) activated, the response can depend on the configuration of the RMM . For more information about setting up the RMM, see the Risk Management Module Guide.
8 ➡	If the authentication response shows that the shopper was authenticated, then Worldpay verifies that the authentication response belongs to the authentication request. If verification is successful, Worldpay proceeds to exchange the authorisation information with the acquirer, including the 3D Security authentication information.
9 ➡	After receiving an authorisation response from the acquirer, Worldpay sends an AUTHORISED response to the merchant.

Table 11: Key to Figure 4: 3D Secure process flow

7.2 Example initial XML order

The example code below shows the initial XML order sent by the merchant to the Worldpay payment service (see step / arrow 2 in **Figure 3: 3D Secure process flow**).

```
<?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="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>3D</cardHolderName>
            </VISA-SSL>
        <session shopperIPAddress="123.123.123.123" id="021ui8ib1"/>
        </paymentDetails>

        <shopper> [example using Firefox 3.5.5 to demonstrate]
    <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></order></submit>
</paymentService>
```

Code example 20: Initial XML message in 3D Secure process flow

Example notes:

<cardHolderName> When you send an initial XML order to the Worldpay payment service test environment, the `cardHolderName` element must contain "3D" as the card holder name.

<browser>
<acceptHeader>
<userAgentHeader> The `browser`, `acceptHeader` and `userAgentHeader` elements must not be hard coded by your system.

For more information about structuring an XML order, see ***4 Structure of an XML Direct order.***

7.3 Example reply to initial XML order message

The example code below shows the reply sent by the Worldpay payment service to the initial XML order (see step / arrow 3 in **Figure 3: 3D Secure process flow**).

```
<?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://securetest.worldpay.com/jsp/test/shopper/VbV_TestIss
uer.jsp]]>
</issuerURL>
          </request3DSecure>
          </requestInfo>
              <echoData>-1374244409987691395</echoData>
</orderStatus></reply>

</paymentService>
```

Code example 21: Example reply to initial XML order message

Example notes:

<requestInfo> This message extends the `orderStatus` element with anew sub-element, `requestInfo`. The `requestInfo` element contains requests for information on the submitted order.

<request3DSecure> The `request3DSecure` element contains the request for 3D Secure authentication.

<paRequest> The `paRequest` element contains data that was received from the 3D Security Directory. This data must be supplied as-is in the redirect messageto the issuer's 3D Secure authentication page.

<issuerURL> The `issuerURL` element contains the URL of the 3D Secure authentication page where the shopper is redirected.

<echoData> The `echoData` element is used by Worldpay to process all the following messages, belonging to the same transaction, more efficiently. This element must be supplied in all subsequent messages as-is.

Your system must also extract the session cookie passed back in the HTTP header of this reply message.

*This cookie is returned in the HTTP header of the second XML order message (see step / arrow 6 in **Figure 3: 3D Secure process flow**), which includes the payer authentication response data.*

7.4 Example HTML redirect page

When the merchant receives the first reply with the request for 3D Secure authentication, the merchant must redirect the shopper to the issuer's 3D Secure authentication site (see step/arrow 4 in **Figure 3: 3D Secure process flow**).

You redirect the shopper to the URL of the issuer's 3D Secure authentication siteby submitting an HTTP POST. The HTTP POST:

- Must contain the `PaReq` attribute in the name attribute. The value for the `PaReq` must be the data supplied in the `paRequest` element of the reply message
- Must contain the `TermUrl` attribute. The value for the `TermUrl` is a URL pointing to the merchant's website. This URL specifies where the shopper will be redirected from the issuer's 3D Secure authentication site
  Note: The merchant is responsible for supplying the correct value.
- Must contain the `MD` attribute, although this can optionally be empty. The `MD` attribute:
  - If not empty, must contain only ASCII characters in the range 0x20 to 0x7E; if other data is needed, the field must be Base64 encoded. The size of the field (after Base64 encoding, if applicable) is limited to 1024 bytes. If MD includes confidential data (such as the PAN), it must be encrypted
  - Is supplied in the same form as it is written in the final post when the shopper is redirected from the issuer's 3D Secure authentication site to the merchant's site
  - 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

The URL of the 3D Secure authentication page, to which the HTTP POSTis submitted, is given in the `issuerURL` element of the reply message.

The following example HTML page redirects the shopper to the issuer's 3D Secure authentication site.

Provided that the shopper has enabled Javascript in the browser, the shopper will automatically be forwarded to the Issuer's site. If Javascript has been disabled, the shopper must press the **Submit** button to continue.

```
<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 Submit button. <br/>
After you successfully identify yourself you will be sent back to this website,
where the payment process will continue.<br/>
<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>
```

Code example 22: Example HTML redirect page

7.5 Example second XML order

The second order message is almost the same as the initial order message (see step/arrow 6 in **Figure 3: 3D Secure process flow**). Only two elements are added:

- The `info3DSecure` element (and sub elements)
- The `echoData` element

```
<?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="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>3D</cardHolderName>
            </VISA-SSL>
        <session  shopperIPAddress="123.123.123.123"  id="021ui8ib1"/>
        <info3DSecure>
               <paResponse>somedata</paResponse>
         </info3DSecure>
        </paymentDetails>

        <shopper>  [example  using  Firefox  3.5.5  to  demonstrate]
    <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>1374244409987691395</echoData>
        </order>
          </submit>
              </paymentService>
```

Code example 23: second XML order

Example notes:

`<Info3DSecure>` The `info3DSecure` element contains the 3D Secure authentication response data received by the shopper and the merchant by the issuer.

`<echoData>` You must supply the same data in the `echoData` element as you received in the first reply message from the Worldpay payment service.

Your system must:

- Ensure that there are no differences in the second XML order message, other than the additional elements (highlighted in red) shown in the example. If other changes are made, the second order message will be rejected by the Worldpay system
- Return the session cookie extracted from the HTTP header of the initial XML order reply message (step/arrow 3 in **Figure 3: 3D Secure process flow**) in the HTTP header of the second XML order message (step/arrow 6 in **Figure 3: 3D Secure process flow**). The session cookie is case sensitive

If Javascript has been disabled, the shopper is provided with a link / button that enables them to continue to 3D Secure authentication.

7.5.1 Second XML order reply message

The reply to the second XML order reply from Worldpay informs you if payment has been AUTHORISED by the acquirer or REFUSED (see step/arrow 9 in **Figure 3: 3D Secure process flow**).

If a shopper fails to authenticate themselves successfully, you are sent a REFUSED reply.

If the shopper authenticates successfully, you are sent an AUTHORISED reply.

For more information about responses to an XML order, see **5 Responses to an XML Direct order**.

8 Submitting a MasterPass order

MasterPass is a secure digital wallet service provided by participating banks and supported by Mastercard.

The digital wallet makes online shopping secure and simpleand removes the need for shoppers to share card details.

MasterPass has the following benefits:

For shoppers, MasterPass saves time and provides an additional layer of security for their card details.

For you the merchant, MasterPass generates a higher rate of completed transactions.

8.1 Enabling MasterPass payments

The MasterPass payment method is implemented differently to most payment methods in the XML Direct model. You must:

- Redirect the shopper to MasterPass to allow the shopper to authenticate with MasterPass and agree to the payment
- Use the identifier MASTERPASS-SSL in your initial XML request, and include three URLs (covered below)
- Be aware that the billing address stored by MasterPass overrides any billing address submitted by you, for the purpose of checking

For more information about MasterPass, see: **https://masterpass.com/**.

8.2 Structuring the MasterPass order

The MasterPass payment method is implemented differently to most payment methods in the XML Direct model because you are required to redirect the shopper to MasterPass to allow the shopper to authenticate with MasterPass and agree to the payment.

The initial XML request contains the order, including the payment method details. The identifier for the MasterPass payment method, specified in the `paymentDetails` element, is MASTERPASS-SSL.

You must include three URLs in the order:

Element	Description
successfulURL	The URL where the shopper is redirected upon successfully completing the MasterPass payment. To help you find a successful order, you can choose a unique URL for each transaction.
failureURL	The URL where the shopper is redirected if the MasterPass payment is not successful. To help you find a failed transaction, you can choose a unique URL for each transaction.
cancelURL	The URL where the shopper is redirected if the Cancel or Back to merchant link is clicked on the MasterPass pages. This URL can be made unique for each transaction.

Table 12: URL elements in a MasterPass order

*To avoid an error, do **not** include the session element (for example,* `<session shopperIPAddress="192.123.12.11" id="session12345"/>` *) in MasterPass payment requests.*

8.2.1 MasterPass paymentDetails

The following example shows the minimum paymentDetails required for a MasterPass payment:

```
<paymentDetails>
    <MASTERPASS-SSL>

<successURL>http://worldpay.com/masterpass/masterpass_success.html</successURL>

<failureURL>http://worldpay.com/masterpass/masterpass_failure.html</failureURL>

<cancelURL>http://worldpay.com/masterpass/masterpass_cancel.html</cancelURL>
    </MASTERPASS-SSL>
</paymentDetails>
```

Code example 24: paymentDetails: MasterPass payment

Example notes:

The content is highlighted in red.

`<MASTERPASS-SSL>` The payment method code MASTERPASS-SSL is used for the MasterPass digital wallet service.

`<URL>` The URL that relates to a success, failure or cancel outcome.

Merchants have no control over which payment methods are displayed in the MasterPass wallet.

8.2.2 MasterPass billing address priority

For MasterPass transactions, the billing address stored with MasterPass (entered by the cardholder) takes precedence over any billing address submitted by you.

Any billing address you supply for MasterPass transactions is overridden by the address stored by Mastercard. The Address Verification Service (AVS) also checks the address supplied by MasterPass, **not** the address submitted by you.

8.2.3 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>
```

Code example 25: shopperEmailAddress: MasterPass payment

Example notes:

The content is highlighted in red.

When a payment has been made with the digital wallet, MasterPass sends its own email confirmation to the customer. To send a Worldpay confirmation email to your customer, you must provide a shopper email address in the transaction details passed to the Worldpay payment page.

8.2.4 Setting the shopper language

Because the MasterPass payment method requires interaction with the shopper, you may want to determine the language of the MasterPass login screen that appears to the shopper.

To control the language that appears, include a `shopperLanguageCode` attribute in the order tag of the initial XML message, as shown below:

```
<order  orderCode="masterpasstestorder123"  shopperLanguageCode="en">
```

Code example 26: shopperLanguageCode: MasterPass payment

Example notes:

The content is highlighted in red.

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

8.3 MasterPass responses

When the payment service receives an order request for a MasterPass payment, the payment service attempts to place the order request with MasterPass. This section describes how the payment service responds to your system, depending on whether the order request was successful with MasterPass or unsuccessful.

8.3.1 MasterPass successfully receives the order request

If MasterPass successfully receives the order request, the payment service produces a response and sends it to your system. It should be similar to code example 34 below:

```
<?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="masterpassexampleorder123">
      <reference id="2300036716">
<![CDATA[https://masterpass.com/Checkout/Authorize?oauth_token=0bc0b73fd9085afc2ea62c9d62
7e533e602cf3ff&acceptable_cards=maestro,visa,master&checkout_identifier=a466w4wyhku6khvsd
8acv1hw0ihxda37fn&version=v4&suppress_shipping_address=true&auth_level=basic]]>
      </reference>
    </orderStatus>
  </reply>
</paymentService>
```

Code example 27: Success response: MasterPass payment

The reply includes the **order code**, a unique numeric reference to the order, and a **redirection URL for MasterPass**. It is up to you to redirect the shopper to this URL. This URL causes the shopper to be automatically taken to the MasterPass payment page. Once at this page, the shopper will login to their eWallet and make the payment.

The ampersands in URLs are escaped with SGML entities to allow them to be included in XML messages.

8.3.2 Shopper successfully completes their payment

When a successful payment is made through the MasterPass wallet, the shopper is returned to your successURL, passed in your original XML order request.

To verify that the payment was authorised, use the authorised notification response. For more information, see the **Payment Notifications Guide**.

8.3.3 MasterPass is unable to receive the order request

If there is a problem with the MasterPass service, the response from the payment serviceto your system will look similar to code example 35 below:

```
<?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="masterpassexampleorder123">
     <error  code="7"><![CDATA[Gateway  Error:  Cannot  initialise  masterpass
payment]]></error>
   </orderStatus>
 </reply>
</paymentService>
```

Code example 28: Error response: MasterPass payment

Example notes:

The error in the example occurs as a result of a failed connection with MasterPass.

9 Receiving AAV data

The **American Express Advanced Verification (AAV)** service was implemented by American Express in March 2013.

When a shopper uses an Amex card to make a purchase:

- The AAV service checks the cardholder name, telephone number and email address that the shopper enters against the details held by American Express
- American Express sends the result of these checks (where applicable) to Worldpay

9.1 Enabling AAV

By default, AAV checks are disabled in the Risk Management Service. To enable AAV checks, you must:

- Configure your system to receive the new values generated by the checks in your XML response
- Ensure that you capture the cardholder name, email address and telephone number on your payment pages
- Send the data required by the AAV service as part of your authorisation request

When you have configured your system to receive AAV data, you can enable AAV checks in the Risk Management Service by emailing support@worldpay.com.

9.2 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 (for example, SHOPPER DATA MATCHES)
- As a security-level single character value (for example, A)

9.2.1 Receiving AAV data as a descriptor

The AVV descriptor values returned by Worldpay are shown in the following table:

Value	Description
SHOPPER DATA MATCHES	The data entered by the shopper matches the data held by American Express for the Amex card.
SHOPPER DATA DOES NOT MATCH	The data entered by the shopper does not match the data held by American Express for the Amex card.

Value	Description
DATA NOT SENT	The data (either the cardholder name, telephone number or email address) was not received by American Express. The shopper may not have entered the data.
DATA NOT CHECKED BY ACQUIRER	American Express has not checked the data (either the cardholder name, telephone number or email address).
UNKNOWN	The AAV check was not carried out foran unknown reason (for example, a technical error).

Table 13: AAV descriptor values

9.2.2 Example XML response with AAV descriptors

The following example shows an XML response with AAV data sent as descriptors:

```
<?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>
```

Code example 29: Example XML response with AVV data sent as descriptors

Example note:

The AAV content is highlighted in red.

9.2.3 Receiving AAV data as a security-level single character value

The AAV single character values returned by Worldpay are shown in the following table:

Value	Description
A	Data matched. The data entered by the shopper matches the data held by American Express for the Amex card.
B	Data not checked. American Express has not checked the data (either the cardholder name, telephone number or email address).
C	Data not supplied. The data (either the cardholder name, telephone number or email address) was not received by American Express. The shopper may not have entered the data.
D	Data not matched. The data (either the cardholder name, telephone number or email address). entered by the shopper does not match the data held by American Express for the Amex card.

Table 14: AAV single character values

9.2.4 Example XML response with AAV data sent as single character values

The following example shows an XML response with AAV data sent as single character values:

```
<?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"/>
              <AAVEmailResultCode description="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="22" second="1"/>
        </orderStatus>
      </reply>
    </paymentService>
```

Code example 30: Example XML response with AVV data sent as single character values

Example note:

The AAV content is highlighted in red.

10 Testing in the XML Direct model

This chapter provides guidance on in the XML Direct model.

It tells you how to test:

- Your connection with the Worldpay payment service
- XML Direct orders, including 3D Secure orders

10.1 Test environment

Worldpay provides atest environment for submitting test XML Direct messages at **https://secure-test.worldpay.com/jsp/merchant/xml/paymentService.jsp**.

Before submitting XML messages to the test environment, check that:

- The HTTPS content type is “text/xml”
- The content length is specified correctly. You will not create errors if you do not specify the content length, but you will create errors if you specify the length incorrectly

Before you can submit XML messages, the test environment must be activated for your account by Worldpay. For more information, contact **support@worldpay.com**.

The Worldpay payment service only accepts incoming XML messages if the originating IP address is registered for the merchant.

For more information about registering and managing multiple IP address ranges, see the ***Merchant Interface Guide***.

10.1.1Testing 3D Secure orders: test and production environments

There are some important differences between the test environment and the production environment. These differences are particularly important if you are testing 3D Secure orders.

Element	Comments
issuerURL	The `issuerURL` element in the test environment contains no parameters: **http://example.issuer.url/3dsec.html** However, in the production environment this URL would normally have parameters in place. For example: **https://example.issuer.url/pa.jsp?partner=m&CAA=B** The `PaReq`, `TermUrl` and `MD` elements must be posted with these parameters. **Note:** The redirect to the `issuerURL` must always be made with a POST and not a GET.

Element	Comments
paResponse	The acceptable values for `paResponse` in the Test environment (IDENTIFIED or NOT_IDENTIFIED) are significantly shorter than the values returned from the issuerin production, where the typical length can be up to 4.7 KB. The `paResponse` values: • Must be collected by the merchant system for sending to Worldpay in the second order message • Require extra storage space **Note:** Do not submit a long (4.7KB) `paResponse` to the test environment, as this causes a parsing error.
PaReq	The value of the `PaReq` attribute must be URL encoded before transmission to the issuer.

Table 15: 3D Secure: testing and production environments

10.2 Test values

You can simulate different outcomes when submitting XML Direct orders by entering the following values as the cardholder name (`<cardHolderName>`):

Value	Description
REFUSED	Simulates a REFUSED payment.
REFERRED	Simulates a refusal with the refusal reason REFERRED.
ERROR	Simulates a payment that ends in an ERROR.

Table 16: Test values

10.3 Test credit and debit card numbers

To help you test your system, Worldpay provides aset of test credit and debit card numbers.

For the list of test credit and debit card numbers, see **Appendix F: Test card numbers**.

10.4 Testing Captures and Refunds

You can simulate Captures and Refunds:

- In the **Merchant Interface > Payment and Order Details** by using the **Capture** or **Refund** button
- In the **test environment** by sending an XML Capture or Refund order modification

10.5 Testing 3D Secure orders

To help you test 3D Secure orders, Worldpay provides a dummy card issuer site. The value of the cardHolderName element in the XML order message can be used to simulate various outcomes, as shown in **Table 17: 3D Secure testing: cardHolderName value**.

Before you can test 3D Secure orders your Worldpay account must be enabled for 3D Secure. To enable 3D Secure, contact **support@worldpay.com**.

CardHolderName value	Test environment behaviour
3D	The payment card is participating in 3D Secure. The simulator authentication page is initiated, where you can select further options.
NO3D	The payment card is not participating in 3D Secure. The simulator authentication page is not initiated. The 3DS Result is **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 authentication page is not initiated. The 3DS Result is **Authentication Unavailable**.
Any other value	Any other value initiates a normal, non-3D Secure transaction process.

Table 17: 3D Secure testing: cardHolderName value

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 menu:

paResponse value	Outcome
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, and the order proceeds to authorisation.

paResponse value	Outcome
ERROR 3DS_INVALID_ERROR_CODE	Response failed validation checks The order does not proceed to authorisation.

Table 18: 3D Secure testing: paResponse value

Appendix A: Payment method codes

To determine which payment methods the shopper can use, the merchant can use either:

- The `paymentMethodMask` variable
- The `preferredPaymentMethod` variable

The payment method codes are shown in the tables below.

For the full list of payment methods, see the Worldpay DTD at **http://dtd.worldpay.com/v1/**. *For more information about the alternative payment methods supported by us, see the* ***Alternative Payment Methods Guide***.

Credit and debit cards

Payment method	Payment method code	Area	Comments
American Express SSL	AMEX-SSL	International	-
Visa	VISA-SSL	International	Visa Credit/Debit/Electron
Mastercard	ECMC-SSL	International	The name Eurocard is no longer in use.
AirPlus	AIRPLUS-SSL	International	-
Aurore	AURORE-SSL	International	-
Carte Bancaire	CB-SSL	France	-
Carte Bleue	CARTEBLEUE-SSL	France	-
Dankort	DANKORT-SSL	Denmark	-
Diners	DINERS-SSL	International	-
Discover Card	DISCOVER-SSL	United States	-
GE Capital	GECAPITAL-SSL	International	-
Maestro	MAESTRO-SSL	International	-
Japanese Credit Bank	JCB-SSL	International	-

Payment method	Payment method code	Area	Comments
Laser Card	LASER-SSL	Ireland	-
PayPal	PAYPAL-EXPRESS	International	Card/eWallet
UATP	UATP-SSL	International	-

Table 19: Credit and debit cards

Online debit methods

Payment method	Payment method code	Area	Comments
Electronisches Lastchriftverhfahren	ELV-SSL	Germany	-
Maestro	MAESTRO-SSL	UK	Depending upon the issuer policy, you may need to include either the issuernumberorthe start date in the paymentDetails. See **4.3.5 paymentDetails child element**.

Table 20: Online debit methods

Offline payment methods

Payment method	Payment method code	Area	Comments
Direct bank transfer Redirect bank transfer	TRANSFER_AT-BANK TRANSFER_BE-BANK	Austria 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	

Payment method	Payment method code	Area	Comments
	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	UK	
Giropay	GIROPAY-SSL	Germany	-
Signed Direct Debit	PERMANENT_SIGNED_DD	Germany, Netherlands, Spain and USA	-
Unsigned Direct Debit	SINGLE_UNSIGNED_DD	Germany, Netherlands, Spain and USA	-

Table 21: Offline payment methods

Appendix B: ISO currency codes

The currencies accepted by the Worldpay payment service are listed in the table below.

For the full ISO 4217 list of ISO currency codes, see **http://www.iso.org**.

Worldpay does not take responsibility for an external link's operation or content.

The values in the orders sent to Worldpay use ***exponent*** *instead of* ***decimal*** *delimiters. The currency code is always presented in capitals. For more information, see* **4 Structure of an XML Direct order**.

ISO currency codes

Currency	ISO currency code	Exponent
Nuevo Argentine Peso	ARS	2
Australian Dollar	AUD	2
Brazilian Real	BRL	2
Canadian Dollar	CAD	2
Swiss Franc	CHF	2
Chilean Peso	CLP	0
Yuan Renmimbi	CNY	2
Colombian Peso	COP	2
Czech Koruna	CZK	2
Danish Krone	DKK	2
Euro	EUR	2
Pound Sterling	GBP	2
Hong Kong Dollar	HKD	2
Hungarian Forint	HUF	2

Currency	ISO currency code	Exponent
Indonesian Rupiah	IDR	2
Iceland Krona	ISK	0
Japanese Yen	JPY	0
Kenyan Shilling	KES	2
South Korean Won	KRW	0
Mexican Peso	MXN	2
Malaysian Ringgit	MYR	2
Norwegian Krone	NOK	2
New Zealand Dollar	NZD	2
Philippine Peso	PHP	2
New Polish Zloty	PLN	2
Swedish Krone	SEK	2
Singapore Dollar	SGD	2
Thai Baht	THB	2
New Taiwan Dollar	TWD	2
US Dollar	USD	2
Vietnamese New Dong	VND	0
South African Rand	ZAR	2

Table 22: ISO currency codes

Appendix C: ISO country codes

The countryCode element is used in XML messages.

The country code is an upper-case two letter ISO 3166 standard country code, as shown in the following example:

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

Code example 31: countryCode

For the full ISO 4217 list of ISO country codes, see **http://www.iso.org***.*
Worldpay does not take responsibility for an external link's operation or content.

For more information about structuring XML messages, including address information, see **4 Structure of an XML Direct order***.*

Appendix D: Acquirer response codes

Worldpay uses ISO 8583 response codes in `orderStatusEvent` messages to show you the status of a payment (for example, AUTHORISED or REFUSED).

The response codes (including their numeric value and their mapping to a status) are listed in the table below.

*For more information about the different payment statuses that a payment can obtain during its life cycle, see the **Payment Status Definitions Guide**.*

For more information about responses to XML orders, see **5 Responses to an XML Direct order**.

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

Card message value	Status	Code message value	Status
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	-	-

Table 23: ISO 5853 response codes

Appendix E: CVC/CVV and AVS

Security Code (CVC/CVV) and Address Verification (AVS) checks help you to authenticate a transaction by comparing information entered by the shopper during the payment process with details held by the card issuer.

You can carry out CVC/CVV and AVS checks on an XML Direct order. The example below shows an example of a CVC coded fragment of an XML Direct Order:

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

Code example 32: CVC coded fragment

For more information about structuring an XML Direct order, see **4 Structure of an XML Direct order***.*

The Worldpay payment service only carries out CVC/CVV and AVS checks on valid XML code.

Testing CVC/CVV

You can simulate the outcome of CVC/CVV checks using the codes in the table below:

CVC/CVV code	Simulated scenario
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

Table 24: Testing CVC/CVV

You can simulate the outcome of CVC/CVV checks for American Express, using the codes in the table below:

CVC/CVV code	Simulated scenario
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

Table 25: Testing CVC/CVV for American Express

Testing AVS

You can simulate the outcome of AVS checks (on the billing address), using the codes in the table below:

AVS code	Simulated scenario
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

Table 26: Testing AVS

Appendix F: Test card numbers

You can use the following credit / debit 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.

For more information about testing your XML Direct integration, see **10 Testing in the XML Direct model**.

Test card numbers

Card type	Test card number
Airplus	122000000000003
American Express	343434343434343
Cartebleue	5555555555554444
Dankort	5019717010103742
Diners	36700102000000 and 36148900647913
Discover card	6011000400000000
JCB	3528000700000000
Laser	630495060000000000 and 630490017740292441
Maestro	6759649826438453 and 6799990100000000019
Mastercard	5555555555554444 and 5454545454545454
Visa	4444333322221111, 4911830000000 and 4917610000000000
Visa Debit	4462030000000000 and 4917610000000000003
Visa Electron (UK only)	4917300800000000
Visa Purchasing	4484070000000000

Table 27: Test card numbers

Appendix G: XML error codes

The list of XML error codes is as follows:

1. Internal error, a general error

2. Parse error, invalid XML

3. Security error

4. Invalid request

5. Payment details in the order element are incorrect.

6. 3D Secure error: Could not find bean(s) in session cache

Example: 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="WPACC11112222">
       <reply>
          <error code="1"><![CDATA[IInternal error]]></error>
       </reply>
     </paymentService>
```

Code example 33: Error code 1. Internal error, a general error

Example note:

The error code is highlighted in red.

Internal errors originate with the Worldpay payment service, and are usually addressed quickly. If you encounter an internal error, we recommend that you try submitting the XML message again after a brief period.

Example: Error code 2. Parse error, invalid XML

XML message posted empty

```
<?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="WPACC11112222">
      <reply>
         <error code="2"><![CDATA[Empty body in message]]></error>
      </reply>
    </paymentService>
```

Code example 34: Error code 2. Parse error, invalid XML

Example note:

The error code is highlighted in red.

The error above indicates that the body of the XML message posted was empty. This error is also returned when the content length has been set incorrectly (too few characters have been specified).

Incorrect / missing DOCTYPE declaration

```
<?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="WPACC11112222">
        <reply>
          <error code="2"><![CDATA[Missing DOCTYPE declaration]]></error>
       </reply>
     </paymentService>
```

Code example 35: Error code 2. Parse error, invalid XML

Example note:

The error code is highlighted in red.

The example error above indicates that the XML code sent to Worldpay does not contain the required doctype declaration. This is used by our payment service to determinewhat kind of information is being sent.

Example: 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="WPACC11112222">
         <reply>
            <error code="4"><![CDATA[Security Violation.]]></error>
        </reply>
</paymentService>
```

Code example 36: Error code 4. Security error

Example note:

The error code is highlighted in red.

The error code above usually indicates one of the following:

- There is a difference between the Merchant Code used to set up the connection and that referred to in the XML message
- A connection has been attempted from an unregistered IP
- The merchant is submitting to an inactive environment (usually because they have only activated the Test environment, and are attempting to submit to production)

Example: 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="WPACC11112222">
       <reply>
         <orderStatus orderCode="123456">
             <error code="5"><![CDATA[Duplicate Order]]></error>
        </orderStatus>
      </reply>
    </paymentService>
```

Code example 37: Error code 5. Invalid request

Example note:

The error code is highlighted in red.

Each `orderCode` has to be unique. In the example above 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 an `orderCode` unique is to use a date/time-stamp, an incremental number or a combination of both.

Example: 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="WPACC11112222">
         <reply>
           <orderStatus orderCode="1112">
             <error code="7"><![CDATA[Invalid payment details : Expiry date =
01/2002]]></error>
           </orderStatus>
        </reply>
      </paymentService>
```

Code example 38: Error code 7. Payment details in the order element are incorrect

Example note:

The error code is highlighted in red.

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

Example: Error Code 7. 3D Secure error. Could not find bean(s) in session cache

```
<?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[Internal error! Could not find bean(s) in
session cache.]]></error>
          </orderStatus>
        </reply>
      </paymentService>
```

Code example 39: Error code 7. Could not find bean(s) in session cache

Example note:

The error code is highlighted in red. This payment has been refused because the cookie reference was not supplied by your system fora 3D secure payment.

For more information about the correct way to send cookie references, see: **7.3 Example reply to initial XML order message** and **7.5 Example second XML order**.

Appendix H: Revisions to the guide

Revision	Release date	Changes
6.5	June 2019	Updated 3D Secure sections: 3D Secure is now a mandatory scheme.
6.4	July 2016	Updated orderCode attribute details.
6.3	March 2016	**Removed:** ● V.me has been removed from the guide because the V.me (by VISA) product has been withdrawn
6.2	March 2015	**Added:** ● MasterPass information has been rewritten for clarity in sections 8.3, 8.3.1, 8.3.2and 8.3.3 ● American Express test card number modified in Appendix F: Test card numbers ● The table of revisions to the guide is now in this appendix.
6.1	February 2015	**Changes** to section 7.4 concerning the MD attribute.
6.0	January 2015	**Added:** ● Example error code 7: Could not find bean(s) in session cache to **Appendix G: XML error codes** **Removed:** ● Online Alternative Payment Methods table (instead referring to the Alternative Payment Methods guide)
5.9	September 2014	**Added:** ● MasterPass information (Section 4.3.5 and a new Section 8)
5.8	August 2014	**Updated:** ● Connecting Using HTTPS (Section 3.1) **Added:** ● A note on how shoppers should have cookies enabled on theirweb browsers. (Section 4.3)
5.7	June 2014	**Updated:** ● Information about submitting a 3D Secure order—removed ref to J/Secure

Revision	Release date	Changes
5.6	May 2014	**Updated:** - List of ISO currency codes and exponents - Minor reword for HTTPS connection
5.5	May 2014	**Updated** - Applied new template
5.4	April 2014	**Updated:** - Information about MCC 6012 Merchants and VISA
5.3	January 2014	**Updated:** - Information about submitting a 3D Secure order
5.2	December 2013	**Added:** - Information about submitting batch orders **Updated:** - Order code examples
5.1	November 2013	**Updated:** - Guide rewritten and restructured - New template applied
5.0	June 2013	**Added:** - Information about the V.me by Visa digital wallet service - Information about American Express Advanced Verification (AAV)
4.7	December 2012	**Updated:** - Information about alternative payment methods was moved to the Alternative Payment Methods Guide
4.6	September 2012	**Updated:** - List of alternative payment methods - Maximum and minimum amounts **Added:** - Information about mandatory and optional fields for alternative payment methods - Information about transaction statuses returned in pendingURL

Revision	Release date	Changes
4.5	July 2012	**Updated:** • List of alternative payment methods **Added:** • Maximum and minimum amounts for alternative payment methods
4.4	June 2012	**Added:** • Code examples for alternative payment methods
4.3	May 2012	**Corrected:** • Payment method code for Yandex.Money

Table 28: Revisions to the guide from May 2012 onwards

Contact us

Support: +44 (0) 870 3661233

UK Sales: 0845 3016251 International Sales: +44 (0)1268 500612

Email: support@worldpay.com

Worldpay Support Centre: http://www.worldpay.com/support/bg