



Hosted Payment Page (HTML Redirect) Guide

Version 2 – April 2009

Table Of Contents

About this Guide.....	5
History	5
Copyright.....	5
Introduction	6
What is Hosted Payment Page (HTML Redirect) Service?	6
How the Hosted Payment Page (HTML Redirect) Works.....	7
MD5 Security.....	8
Getting Started	8
The Payment Pages	9
Running in the Test Environment.....	10
Configuring Your Installation.....	11
Different Currencies	11
Specifying Preferred Merchant Codes (Accounts).....	12
Hosted Payment Page (HTML Redirect) Example Sites	12
Payment Notifications (Callbacks)	13
A Simple Integration.....	14
Purpose.....	14
Use.....	14
Setting-up Quickly.....	14
Developing Further and Going Live	18
The Order Details Submission	19
Purpose.....	19
Use.....	19
Order Details Form Structure.....	19
URL Order Details String	21
Optionally Complex	22
Order Details Parameters	23
Example Order Details Form with Data	24

Passing Fixed Data.....	25
A Complete HTML Page.....	26
Test Card Numbers.....	30
Payment Result - to Shopper.....	32
Purpose.....	32
Use.....	32
Shopper's Result Pages.....	32
Uploading Result Pages.....	33
Payment Result - to You.....	35
Purpose.....	35
Use.....	35
Error Messages and Problems.....	37
Introduction.....	37
I Copied the HTML Example Code and It Doesn't Work.....	37
Merchant Has No Suitable Accounts for This Purchase.....	37
I Get a Processing Error When I Attempt a Transaction.....	38
Some of My Chosen Currencies are Missing.....	38
Redirect Model Parameters.....	39
Purpose.....	39
Use.....	39
Mandatory Parameters.....	39
Initiating Purchase Transaction Parameters.....	40
Payment Page Parameters.....	45
Payment Result Parameters.....	46
Recurring Payments (FuturePay) Parameters.....	48
Custom Parameters.....	49
Tags Used in Substitution.....	50
Purpose.....	50
Use.....	50
Parameter and File Substitution.....	50

Hosted Payment Page service

Enhancing Security	53
Purpose	53
Use	53
What is MD5?.....	53
Implementation and Prerequisites	54
Additional Security	56
Appendices	57
Introduction	57
Introduction	57
Exchange Rates.....	57
ISO Currency Codes.....	59
ISO Country Codes.....	61

About this Guide

Welcome to the Submitting Transactions in the Hosted Payment Page (HTML Redirect) Guide. This guide describes how to connect your Internet shop to the RBS WorldPay payment service, using our HTML integration method. It provides practical guidance as well as in-depth reference material.

To get the most from this guide, you will need to have a working knowledge of HTML, including forms. However, please note that you can get going with your Internet shop very quickly without being an HTML expert, please refer to [The Easiest Option](#) for details.



The term 'merchant' refers to a business that is using RBS WorldPay payment service to process payments. The term 'shopper' refers to the purchaser of a merchant's products and services.

History

Change description	Date	Affected Pages
RBS WorldPay rebrand	February 2009	All pages

Copyright

RBS WorldPay © The Royal Bank of Scotland

While every effort has been made to ensure the accuracy of the information contained in this publication, the information is supplied without representation or warranty of any kind, is subject to change without notice and does not represent a commitment on the part of The Royal Bank of Scotland. The Royal Bank of Scotland, therefore, assumes no responsibility and shall have no liability, consequential or otherwise, of any kind arising from this material or any part thereof, or any supplementary materials subsequently issued by The Royal Bank of Scotland. The Royal Bank of Scotland has made every effort to ensure the accuracy of this material.

Introduction

What is Hosted Payment Page (HTML Redirect) Service?

The Hosted Payment Page (also known as HTML Redirect) is an integration method to the RBS WorldPay payment service, suited for Internet shop environments, call centers, or reservation centres, multi channel sales situations. It allows for real-time processing of payments and ensures a maximum number of up-to-date payment methods. The Hosted Payment Page service is secure, provides RBS WorldPay with required information to perform active fraud risk assessment, and is the fastest way to get up and running with on-line payments.

This integration method uses HTML messages to pass information between your site and RBS WorldPay. It is the simplest and easiest method of integration and will work on just about any platform.

Nothing to Install

There is nothing to install with the Hosted Payment Page method of integration. All you need is a working Internet connection and your Installation ID.

The Easiest Option - Storebuilders

If you have only a modest understanding of HTML, the easiest way to get your website shop up and running is to use an off-the-shelf storebuilder package that is already integrated with Redirect. For a list of these, please refer to <http://www.rbsworldpay.com/support/bg/index.php?page=development&sub=cart>.

If you are taking this route, please also refer to the [Test and Go Live Guide](#) for information about testing your website and going live with it.

Quick and Simple, or Complex

If you are familiar with HTML you can choose to set-up a simple integration using examples provided in this guide - and to make use of our standard payment pages - all you need to do is to copy a few lines of HTML from examples in this guide and paste them into your website page. For more information, please refer to [A Simple Integration](#).

If you have advanced HTML skills you can develop your own fully automated ordering system, complete with customised payment pages.

Note that you can even integrate using a single URL query string, for details please refer to [URL Order Details String](#).

The choice is yours.

Further Choices

- You can build your store around any shopping cart package already integrated with Redirect. For a list of these, please refer to <http://www.rbsworldpay.com/support/bg/index.php?page=development&sub=cart>
- You can customise our payment pages so that they fit in with the look and feel of your own website. For more information, please refer to [Configuring Your Installation](#).
- You can choose to have transaction details returned to your server and, if required, display your own transaction confirmation message to your shoppers, using a feature called Payment Notifications. Note that you do NOT have to use Payment Notifications. For more information, please refer to [Payment Notifications](#).

How the Hosted Payment Page (HTML Redirect) Works

The Hosted Payment Page (also known as HTML Redirect) service allows your website with its own shopping system to work in conjunction with our payment service.

This integration method uses HTML forms to pass information to the RBS WorldPay payment gateway for payment processing. You create the form in your web page and insert parameters that describe the purchase - this form is known as an order details form.

1. When a shopper is ready to pay for their goods, your website should present them with a button or link which, when selected, submits the order details to our payment page.
2. The shopper will then be taken to our payment pages to enter their payment details, such as credit/debit card details. Your website does not gather card details from the shopper - we do this in our payment pages.
3. If the shopper has chosen to pay by credit/debit card:
 - we forward the shopper's details to the bank, where the shopper's credit worthiness is checked
 - the bank returns an authorised or declined response to us
 - if the payment is declined, the shopper is given two options - to try another means of payment or to cancel the purchase.
4. We then display a result page to the shopper showing the outcome of the payment transaction. We also send them a confirmation email. For more information, please refer to [Payment Result - to Shopper](#).
5. We also inform you about the transaction. How this is done will depend upon how you have configured your options with the Merchant Interface, but at the very least we will send you an email. For more information, please refer to [Payment Result - to You](#).

Order Details and Multiple Items

Please note that a single order details submission must describe a single total purchase rather than a series of items in a shopping cart.

If you want to sell multiple items, you must summarise the total purchase amount and description of the goods on your own site before passing the summarised details through to us.

Encryption and Administration

If required, you can add encryption to the order details submission using the MD5 message-digest algorithm. This protects the purchase information from unauthorised tampering. For more information, please refer to [What is MD5?](#).

MD5 Security

As part of our ongoing commitment to safe online trading, we have introduced MD5 security encryption for merchants using the Hosted Payment Page service. This protects order details from unauthorised tampering.

For more information please refer to [Enhancing Security](#).

Getting Started

Once your application to join RBS WorldPay has been approved, you will be contacted by your local Technical Support department. They will provide you with the following:

1. **Installation ID** - Your Installation ID is issued by the Support Team and should not be confused with the RBS WorldPay Administration Code, which is sent to you as soon as your application to us is provisionally accepted.

The Installation ID is important as it must be included in the order details submissions you send from your website shop to us. For more information, please refer to [A Basic Order Details Form](#).

2. **Two separate URLs for each installation** - These URLs are needed for your order details submissions. One will be the URL to use in order details submissions to the Test Environment and the other will be the URL to use in order details submissions to the Production Environment.
3. **Merchant Interface URL** - The Merchant Interface provides various administrative facilities to help you manage your shop. For example:
 - you can access your merchant code (account) details, statements and transaction information
 - you can perform refunds and post-authorisations (if you have chosen to use the post-authorisation service)
 - you can upload files to use in customising the RBS WorldPay payment pages.

4. **Merchant Interface user name and password** - At the end of the application process, you will be asked to provide your password. We will send you an email with your user name. Use both to login to the Merchant Interface for the first time. We recommend that you change your password immediately. Note that you can change this user name and password as often as you like.

For more information about the Merchant Interface, please refer to the [Merchant Interface Guide](#).

Integrating

You can now begin the process of integrating your website shop with our payment service. Note that there is nothing to install for the Hosted Payment Page service.

Integration is very straightforward - when you can successfully send us the order details of a purchase from your website and get an email response from us, you have integrated your website.

To integrate with RBS WorldPay, all you need to do is:

- gather order details
- establish communications between our server and your website - basically this just means sending a test transaction to us and getting an email response

As soon as you have integrated you can then activate your installation and, subject to some checks, you can go live.

Skills You Will Need

For basic integration you will need to know how to use an HTML editor and have a working knowledge of HTML. However, note that you can set-up a simple integration quickly and easily, please refer to [A Simple Integration](#) for details.

If you do not have these skills, you may prefer to buy an off-the-shelf shopping cart that is already integrated with RBS WorldPay. For a list of these, please refer to <http://www.rbsworldpay.com/support/bg/index.php?page=development&sub=cart>

For more complex integration, such as automated ordering systems or your own bespoke, shopping cart functionality, you will need web development expertise, including server-side scripting languages.

The Payment Pages

Once order details have been submitted by your website and received by our payment service, we will present a series of payment pages to the shopper; it is these pages that gather the payment information needed to process the transaction.

Hosted Payment Page service

- These pages can be customised in many ways to suit your website style and presentation. For comprehensive information on customising your payment pages please refer to the [Customising Guide \(Standard\)](#).
- If you have enabled Payment Notifications, you can control how these pages are displayed by using parameters in your submission. For more information please refer to [Payment Notifications](#).



Note that we gather credit/debit card details in these payment pages, there is no need for you to do this in your website.

Currency Selector Box

If some of your chosen currencies are missing from the Currency Selector Box in the payment page, please check that you are using the correct Installation ID for the transaction. If this is correct, then your currencies may not have been added to our system, please contact us.

For more information on currencies, please refer to [Different Currencies](#).

Running in the Test Environment

You can test your site by running in the Test Environment. This can be very useful when you are developing your website, as it enables you to try out your pages with real, live debit and credit cards without transferring funds.

When you run in the Test Environment you are actually using a complete, working environment, which behaves in exactly the same way as the Production Environment - except that no money is actually transferred when you submit transactions.

Accordingly, you will be able access reports of test transaction details with the Merchant Interface. You can also use HTML Redirect to make tests that do not involve your bank or other financial institution. This will enable you to get a realistic sense of the whole payment processing procedure via RBS WorldPay, without any money ever changing hands.

The test environment can be accessed by going to the left-hand menu of the Merchant Interface .

For more information about testing your installation, please refer to the [Test and Go Live Guide](#).



You can also test your live merchant code (account) - note that this is different to running in the Test Environment - for more information please refer to the [Test and Go Live Guide](#).

Configuring Your Installation

You can set-up the Hosted Payment Page (also known as HTML Redirect) to suit your own requirements by configuring various options. For instance, you can upload your own versions of the payment pages that are displayed to your shoppers, and you can specify that you want to use the Payment Notifications (Callbacks) feature.

Although specific configuration tasks will be described in other sections in this guide, you should note that you configure your installation with the Merchant Interface.

1. Login to the Merchant Interface using the user name and password given to you by Technical Support.
2. When you login, you will be presented with a left-hand Menu, select **Installations**.
3. You will then be shown a tabulated list of all your installations and also access to the **Payment Page Editor**.
4. To configure installation options, select the spanner button in an **Integration Setup** column for the installation you are interested in. If you are setting-up for the first time you should initially select the Test column.
5. To customise payment pages, select the **Edit Payment Pages** button. But note that if you have the Payment Notifications feature enabled, you can control how the payment pages are displayed by using parameters. For more information about this advanced topic, please refer to [Display Parameters](#).



Note that there are Test and Production columns for each installation listed. If you are setting-up for the first time you should initially select the Test column, this will enable you to develop and test every feature of your website pages before moving to a real, production environment.

For general information about configuring your installation, please refer to the [Merchant Administration Interface Guide](#), and for information about customising your payment pages, refer to the Customising Guides ([Standard](#) and [Advanced](#)).

Different Currencies

When you apply for a RBS WorldPay merchant code (account), you get at least one free currency. You can add more currencies or currency combinations - where the acquisition and settlement currencies are different - by paying a one-off fee for each currency/currency combination. Please contact us for details.

If you are dealing with a number of currencies on your website you may want to ensure that you are using up-to-date exchange rates. The exchange rates we use are updated every 24 hours at midnight GMT. These rates are then guaranteed for the next 24 hours for all transactions through RBS WorldPay.

You too can use these exchange rates to keep your own website up-to-date. You can obtain a file containing the exchange rates that are valid for the current day. Please refer to [Exchange Rates](#) for further details.

Specifying Preferred Merchant Codes (Accounts)

As a general rule, we open one merchant code (account) per currency set that you process. However, you may need to consider using **preferred** merchant codes if you have a number of merchant codes with identical characteristics but where they will be used for different purposes.

For instance, you may have a merchant code for software sales and another for hardware sales - so, order details submitted to us for software will need to specify the software merchant code, and order details submitted for hardware will need to specify the hardware merchant code.

You specify the merchant code to be used by using the parameter `accId1`. For example:

```
<input type=hidden name="accId1" value="MYCOMPANY">
```

By default, our system searches your merchant codes in order until one suitable for the payment is found.

If none of the specified merchant codes matches, then our system goes on to try any other merchant codes you may have. Hence these parameters specify a preference rather than an absolute list of merchant codes to use.

Hosted Payment Page (HTML Redirect) Example Sites

If you would like to view integration examples prior to building your own, or if you just need some pointers about how to pass the information you gather to us, please visit the following URL:

```
http://www.rbsworldpay.com/support/bg/index.php?page=development&sub=integration&subsub=example
```

This site has links to a number of model shops that the UK Technical Support team have built to assist you.



Note that we cannot accept responsibility for the smooth running of any sites derived from the code on these example pages. The shops exist merely as examples and are not supported RBS WorldPay products.

Payment Notifications (Callbacks)

The Payment Notifications (Callbacks) feature enables you to automate and control your website. For instance, you can send your own result page when your shopper has completed a transaction. You can also benefit from the content of messages we send to you, as follows:

- more data is returned to you in Payment Notifications messages than you would receive in the standard email notification
- you can pass your own parameters through our payment service and back to yourself
- you can confirm the integrity of the data received and processed by us.

For comprehensive coverage of this feature, please refer to the [Payment Notifications Guide](#).

A Simple Integration

Purpose

This section describes what an order details submission is. It describes the structure of a basic order details submission and the use of parameters in it.

Two form-based HTML specimens are provided: one is an order details form that can be pasted into an existing web page and the other is a complete, working web page that will initiate a purchase transaction. Please refer to [Order Details Form Structure](#) and [A Complete HTML Page](#).

When a shopper is ready to pay for their goods, your website should present them with a button or link that, when clicked, submits the order details to our secure server - both of the form-based specimens provide submission buttons.

A URL-based specimen is also provided, where the order details are contained within the URL string.

Use

Please refer to the following sections for details about order details submissions:

- ⇒ [Order Details Form Structure](#) - the structure with mandatory and optional parameters
- ⇒ [URL Order Details String](#) - order details in a query string
- ⇒ [Optionally Complex](#) - you can keep them simple or make them complex
- ⇒ [Order Details Parameters](#) - a brief overview of the parameters
- ⇒ [Example Order Details Form with Data](#) - an example with optional data
- ⇒ [Passing Fixed Data](#) - passing a shopper's contact details to the payment pages
- ⇒ [A Complete HTML Page](#) - a working, integrated web page with submission buttons

Setting-up Quickly

Integration is very straightforward - when you can successfully send us details of an order and get an email response from us, you have integrated your website.

The steps below describe how to set-up an integration using an example HTML form, shown below, which can be used as a template in your own web pages. The steps provide the information you need to get your website integrated quickly and easily.

1. You need to add the example HTML form to the website page that advertises the product you wish to sell.

The form holds the details of the order and it sends this information to us, using its POST method.

2. The HTML code for the form is provided below, simply copy it and paste it into your page, then update the parameter values and elements to suit the product details in your website page.

Please only update the parameter values, not the parameters themselves. If you decide to rewrite the parameters, take care with spelling - they must be exactly as shown below - and note that they are case-sensitive: INSTID is **not** the same as instld.

3. The HTML example code will add a 'Buy This' button to the page.

When you view the page in your browser, select the button and you will be taken to our payment service, you can then follow the progress of the transaction through the payment pages.

4. The HTML example is already set-up for the Test Environment, so you can try it out without worrying about transferring real money.
5. We will send you email notifications of each transaction automatically, including Test Environment transactions.

Your site is integrated if you can successfully get to the payment pages and you can get an email about the test purchase.

A Basic Order Details Form

This basic order details form has a simple structure. It is simply an HTML submission form with a small number of <input> tags containing parameters. The parameters included in this example are mandatory and **must** be included in the form.



Note that the comments inside the <!-- and --> symbols can be included in your HTML page, the browser will ignore them.

```
<!-- The first line of code creates a form, which has the  
POST method and its action is to send the form to us. You do  
not need to set-up a special connection to us before using  
it - your Internet connection is all you need to communicate  
with us.-->
```

```
<form
action="https://secure.wp3.rbsworldpay.com/wcc/purchase"
method=POST>

<!-- This next line contains a mandatory parameter. Put your
Installation ID inside the quotes after value= -->
<input type=hidden name="instId" value="Your installation ID
">

<!-- Another mandatory parameter. Put your own reference
identifier for the item purchased inside the quotes after
value= -->
<input type=hidden name="cartId" value="Your ID for the
product ">

<!-- Another mandatory parameter. Put the total cost of the
item inside the quotes after value= -->
<input type=hidden name="amount" value="The cost of the
product ">

<!-- Another mandatory parameter. Put the code for the
purchase currency inside the quotes after value= -->
<input type=hidden name="currency" value="currency code e.g.
GBP, USD ">

<!-- This creates the button. When it is selected in the
browser, the form submits the purchase details to us. -->
<input type=submit value=" Buy This ">

</form>
```

A Test Version of the Order Details Form

This version of the basic order details form also has a simple structure. It uses the URL for our test environment and includes the `testMode` parameter.



Because this is a test transaction no money ever changes hands. For details of using test submissions please refer to the [Test and Go Live Guide](#).

```
<!-- The first line of code specifies the URL for our test
environment.-->

<form
action="https://secure.wp3.rbsworldpay.com/wcc/purchase"
```

```
method=POST>

<!-- This next line contains the testMode parameter - it
specifies that the submission is a test submission -->
<input type=hidden name="testMode" value="100">

<!-- This next line contains a mandatory parameter. Put your
Installation ID inside the quotes after value= -->
<input type=hidden name="instId" value="Your installation ID
">

<!-- Another mandatory parameter. Put your own reference
identifier for the item purchased inside the quotes after
value= -->
<input type=hidden name="cartId" value="Your ID for the
product ">

<!-- Another mandatory parameter. Put the total cost of the
item inside the quotes after value= -->
<input type=hidden name="amount" value="The cost of the
product ">

<!-- Another mandatory parameter. Put the code for the
purchase currency inside the quotes after value= -->
<input type=hidden name="currency" value="currency code e.g.
GBP, USD ">

<!-- This creates the button. When it is selected in the
browser, the form submits the purchase details to us. -->
<input type=submit value=" Buy This ">

</form>
```



Note that you can specify the type of response you want from our system for a test submission by entering REFUSED, AUTHORISED, ERROR or CAPTURED as the value in the `name` parameter. You can also generate an AUTHORISED response by using a real name, such as, J. Bloggs. For more information please refer to the [Test and Go Live Guide](#).

Developing Further and Going Live

You can use the example order details form as the basis for all your website pages or you can develop them from this point.

When you have completed the development of your pages, and have tested them, you are ready to go live. Please refer to the [Test and Go Live Guide](#) for details.

You can make the example form suitable for the live Production Environment by changing its URL and removing the `testMode` parameter, once again please refer to the [Test and Go Live Guide](#) for details.

Single and Multiple Items

A single order details form must describe a **single** total purchase rather than a series of items in a shopping cart.

If you want to sell multiple items, you must summarise the total purchase amount and the description of the goods on your own site before passing the summarised details through to RBS WorldPay.

The Order Details Submission

Purpose

This section describes what an order details submission is. It describes the structure of a basic order details submission and the use of parameters in it.

Two form-based HTML specimens are provided: one is an order details form that can be pasted into an existing web page and the other is a complete, working web page that will initiate a purchase transaction. Please refer to [Order Details Form Structure](#) and [A Complete HTML Page](#).

When a shopper is ready to pay for their goods, your website should present them with a button or link that, when clicked, submits the order details to our secure server - both of the form-based specimens provide submission buttons.

A URL-based specimen is also provided, where the order details are contained within the URL string.

Use

Please refer to the following sections for details about order details submissions:

- ⇒ [Order Details Form Structure](#) - the structure with mandatory and optional parameters
- ⇒ [URL Order Details String](#) - order details in a query string
- ⇒ [Optionally Complex](#) - you can keep them simple or make them complex
- ⇒ [Order Details Parameters](#) - a brief overview of the parameters
- ⇒ [Example Order Details Form with Data](#) - an example with optional data
- ⇒ [Passing Fixed Data](#) - passing a shopper's contact details to the payment pages
- ⇒ [A Complete HTML Page](#) - a working, integrated web page with submission buttons

Order Details Form Structure

At its most basic an order details submission form has a very simple structure, it is simply an HTML submission form with a small number of <input> tags containing parameters. The parameters hold the details of the order.

The order details form must contain four mandatory parameters, as shown in the HTML example below. It may also contain a range of optional parameters, also as shown below. For more information about parameters, please refer to [Order Details Parameters](#).

Hosted Payment Page service

Your web page or website must also provide a button or link that submits the order details to us when a shopper selects it.

The example form below shows the structure of a basic order details form. Note that it includes a button that will submit the order details to our secure server, using the form's POST method.

You can use this example form in your own website page: simply copy it and paste it into your page, then update the parameters and elements to suit the product details in your page.



Note that an order details form must describe a single total purchase rather than a series of items in a shopping cart. If you want to sell multiple items, you must summarise the total purchase amount and the description of the goods on your own site before passing the summarised details through to us.

The Structure of an Order Details Form

The first four parameters inside the form are mandatory and **must** be included in the order details you send to us.



Note that the comments inside the `<!--` and `-->` symbols can be included in your HTML page, the browser will ignore them.

```
<!-- The first line of code creates a form, it has the POST
method and its action is to send the form to us. You do not
need to set-up a special connection to us before using it -
an Internet connection is all you need to communicate with
us.-->

<form
action="https://secure.wp3.rbsworldpay.com/wcc/purchase"
method=POST>

<!-- A mandatory parameter. Put your Installation ID inside
the quotes after value= -->
<input type=hidden name="instId" value="Your installation ID
">

<!-- A mandatory parameter. Put your own reference
identifier for the item purchased inside the quotes after
value= -->
<input type=hidden name="cartId" value="Your reference ID
for the product ">

<!-- A mandatory parameter. Put the total cost of the item
inside the quotes after value= -->
```

```

<input type=hidden name="amount" value="The cost of the
product ">

<!-- A mandatory parameter. Put the code for the purchase
currency inside the quotes after value= -->
<input type=hidden name="currency" value="currency code e.g.
GBP, USD ">

<!-- An optional parameter. Put your description of the item
inside the quotes after value= -->
<input type=hidden name="desc" value=" what you are selling
">

<!-- An optional parameter. The shopper's email address,
this passes it to our server. We can then send an email to
the shopper to inform them of the transaction. -->
<input type=hidden name="email" value="Shopper's email
address ">

<!-- An optional parameter. This tells our server that the
test transaction is authorised. Note that the shopper's name
parameter is used to specify the test result. -->
<input type=hidden name="name" value="AUTHORISED">

<!-- This creates the button. When it is selected in the
browser the form submits the gathered details to us. -->
<input type=submit value="Buy This">

</form>

```

URL Order Details String

You do not have to use an HTML form to create and submit order details, or to integrate - you can also submit order details in a URL string. For example, you can integrate with this single URL string:

```

https://select-
test.wp3.rbsworldpay.com/wcc/purchase?instId=123456&cartId=R
BSWorldPay+Test&
amount=40.00&currency=GBP&desc=WorldPay+Test&testMode=100

```

As with form-based submissions, the URL order details string must contain the four mandatory parameters, as shown in the example.



Note that the URL above is for the Test Environment and the `testMode` parameter is specified as 100 - will cause our system to treat the submission as a test submission.

You can then specify the type of test result you want by entering an appropriate value in the shopper's name field when the payment page is displayed, you can choose from: AUTHORISED, REFUSED, ERROR, CAPTURED, as shown in the example below. For details about these values, please refer to the [Test and Go Live Guide](#).

```
https://select-  
test.wp3.rbsworldpay.com/wcc/purchase?instId=123456&cartId=R  
BSWorldPay+Test  
&amount=40.00&currency=GBP&desc=WorldPay+Test&testMode=100
```

If you specify any other value in the name field, such as a real shopper's name, the system will respond with an AUTHORISED test result.

The URL string may also contain a range of optional parameters. For more information about parameters, please refer to [Order Details Parameters](#).

Optionally Complex

An order details submission can be simple, where it contains just the four mandatory parameters and a submit button, or it can be complex, where it contains optional parameters that hold shopper details gathered from your own website, as well as parameters that control the appearance of the payment pages.

- With a simple order details submission, where just the mandatory parameters are supplied, the payment pages will open with empty fields - the shopper will then have to enter details, such as name and address, directly into the payment pages.
- Alternatively, if you wish to gather name and address information in your own website pages, you can include this information in your order details submission. We then pass this information on to the payment pages, so that the fields are already populated with this information when the pages are displayed. For more information, please refer to [Passing Fixed Data](#).



You cannot collect card details on your website.

- You can also control the appearance of the payment pages by including parameters for this, please refer to [Display Parameters](#).

Order Details Parameters

An order details submission must contain the four mandatory parameters shown below. It may contain other optional Hosted Payment Page (HTML Redirect) parameters, including custom parameters created by you - it can also contain Recurring Payments (also known as FuturePay) parameters.

This section briefly describes the different kinds of parameters. For more information, please refer to [Hosted Payment Page \(HTML Redirect\) Parameters](#).



Note that all parameter names are case-sensitive.

Mandatory and Optional Parameters

Although there are parameters that are **optional** with regard to the order details submission itself, some of them are **mandatory** with regard to the payment pages. For example, the shopper's address can be optionally included in an order details submission, but if it is not then it **must** be provided by the shopper in the payment pages.

Thus, there are two senses of mandatory parameter:

1. where they are mandatory with regard to the **order details submission**
2. where they are mandatory with regard to the **payment pages**.

Mandatory Order Details Parameters

The following four parameters **must** be included in the order details you submit to us.

<i>parameter</i>	<i>definition</i>	<i>example</i>
instld	Your RBS WorldPay Installation ID.	1234
cartld	A reference you assign to help you identify the purchase.	101KT0098
amount	The total cost of the purchase.	25.35
currency	The purchase currency, please refer to Currency Codes .	GBP

Optional Order Details Parameters

You may include optional parameters when you submit order details, such as a shopper's billing address details. But note that although they are optional in an order details submission, some of these parameters are mandatory in the payment pages, as described above.

You can also include Recurring Payments (FuturePay) parameters in an order details submission. For more information, please refer to [Recurring Payments \(FuturePay\) Parameters](#).

Your Own Custom Parameters

You can create your own custom parameters, to be passed through our payment service and returned to your server in a Payment Notifications message. Custom parameters can also be used in the result pages returned to the shopper in response to their payment.

For more information, please refer to [Custom Parameters](#).

Example Order Details Form with Data

The form-based HTML code shown below is an example of the data that can be submitted to us for a single-item payment.

Note that the example shows billing address details. If you capture this information from the shopper on your website, then it is useful to submit it to us because we automatically place it into billing address fields that the shopper would be required to enter in the payment pages, thus saving them from having to enter it again.



Also note that the shopper can change any of the address details on display in the payment pages. Unless you specify that they cannot, you do this by specifying that these details are fixed data. For more information, please refer to [Passing Fixed Data](#).

For more information about the parameters included in this form, please refer to [Order Details Parameters](#). For a full list of all the parameters that can be submitted to initiate a payment, please refer to [Hosted Payment Page \(HTML Redirect\) Parameters](#).

```
<form action="https://select-  
test.wp3.rbsworldpay.com/wcc/purchase" method=POST>  
<input type=hidden name="testMode" value="100">  
<input type=hidden name="instId" value="1234">  
<input type=hidden name="cartId" value="101KT0098">  
<input type=hidden name="amount" value="25.35">  
<input type=hidden name="currency" value="GBP">  
<input type=hidden name="desc" value="Blue T-Shirt, Medium">  
<input type=hidden name="name" value="J. Bloggs">  
<input type=hidden name="address" value="4 Any St,  
Somewhere">  
<input type=hidden name="postcode" value="AB10 5AB">  
<input type=hidden name="country" value="GB">
```

```
<input type=hidden name="tel" value="0123456789">
<input type=hidden name="email"
value="demo@uk.worldpay.com">
<input type=submit value="Buy This">
</form>
```



Note that in the example above the URL is for the Test Environment and the `testMode` parameter is specified as 100 - this will cause our system to treat the submission as a test submission.

You can specify the type of response you want from our system for a test submission by entering REFUSED, AUTHORISED, ERROR or CAPTURED as the value in the `name` parameter. You can also generate an AUTHORISED response by using a real name, as shown in this example, with J. Bloggs. For more information please refer to the [Test and Go Live Guide](#).

Passing Fixed Data

If you pass the shopper's billing address details to us when you submit order details, we automatically place them into the billing address fields that the shopper would be required to enter in the payment pages. However, the shopper can change these address details in the payment pages unless you specify that they are **fixed data**.

To specify that billing address details are fixed data:

1. make the billing address fields mandatory on your site - how you do this is a matter of personal choice,
2. use an additional parameter in your order details called `fixContact` to lock the contact information in the payment page.

Using the `fixContact` parameter locks the address information passed to us, so that your shoppers cannot change this information when they reach the payment pages, as shown in the example below.

You can also hide the contact details from the shopper when they reach the payment pages. This is done using the `hideContact` parameter, also as shown in the example below.

```
<form action="https://select-
test.wp3.rbsworldpay.com/wcc/purchase" method=POST>
<input type=hidden name="testMode" value="100">
<input type=hidden name="instId" value="1234">
<input type=hidden name="cartId" value="101KT0098">
<input type=hidden name="amount" value="25.35">
<input type=hidden name="currency" value="GBP">
<input type=hidden name="desc" value="Blue T-Shirt, Medium">
<input type=hidden name="testMode" value="100">
<input type="hidden" name="fixContact">
```

```
<input type="hidden" name="hideContact">
<input type=hidden name="name" value="J. Bloggs">
<input type=hidden name="address" value="4 Any St,
Somewhere">
<input type=hidden name="postcode" value="AB10 5AB">
<input type=hidden name="country" value="GB">
<input type=hidden name="tel" value="0123456789">
<input type=hidden name="email"
value="demo@uk.worldpay.com">
<input type=submit value="Buy This">
</form>
```

For more information on parameter names for contact details, please refer to [Hosting Payment Page \(Redirect HTML\) Parameters](#).

A Complete HTML Page

This section describes a simple, but complete and working, integrated HTML page, with an order details submission form already included - and comments.

Using the Page

To use the page, simply copy the entire section of HTML code shown below and save it as an HTML file, then run it in your browser. When the page opens it will display the table shown below.



One-Stop Shop		
Product 1	Price: £5.00	Buy button
Product 2	Price: £10.00	Buy button
Product 3	Price: £15.00	Buy button

Figure: A Complete Page

- Select one of the Buy buttons and then follow the sequence of secure pages to experience a transaction in progress.
- Use the following test card details to proceed with the transaction (for more test card details please refer to [Test Card Numbers](#)):

Visa Card Number: 4917610000000000

or, MasterCard Number: 5100080000000000

Expiry Date: A date later than today but within seven years from submission

Security Code: 123

The transaction will be sent to the Test Environment so no real money will change hands. An email showing the transaction details will be sent to the address you specify in the payment pages.



If you want to view test transaction details for your own merchant code in the Merchant Interface, you will need to enter your own Installation ID in the `instId` parameter value in the HTML code.

HTML for a Complete Page

```
<!-- HTML Redirect Example
0.1
-->
<!-- Please note the code on this pages is provided by RBS
WorldPay as a working example only. -->
<!-- Any changes you make to the copies of these pages will
not be supported by us. -->

<html>
<!-- The name, style, and properties of the page are defined
in between the 'head' tags. -->
<head>
<title>HTML Redirect 0.1</title>
<meta http-equiv="Content-Type" content="text/html">
<meta name="description" content="HTML Redirect Example
0.1">
<meta name="keywords" content="Redirect, html">
<style type="text/css">td {text-align:"left"; vertical-
align:"middle"; font-family:"arial"; color:"black"}
h1,h2,h3,h4,h5,h6,h7 {text-align:"center"; vertical-
align:"middle"; font-family:"arial"; color:"black"}</style>
</head>

<!-- The content to be used on the page is placed between
the 'body' tags. -->
<body>
<!-- This is an order details submission form, for more
information on the elements within a form please refer to
the section Order Details Form Structure. -->

<form action="https://select-
test.wp3.rbsworldpay.com/wcc/purchase" name="BuyForm"
method="POST">
<input type=hidden name="testMode" value="100">
<!-- These first four elements are mandatory. -->
<input type="hidden" name="instId" value="89716">
```

Hosted Payment Page service

```
<input type="hidden" name="cartId" value="abc123">
<input type="hidden" name="currency" value="GBP">
<input type="hidden" name="amount" value="0">

<!-- These elements below are optional. -->
<input type="hidden" name="desc" value="Blue T-Shirt,
Medium">
<input type="hidden" name="name" value="CAPTURED">

<!-- End of order details submission form elements. -->

<!-- JavaScript is used to give functionality to some of the
pages elements. -->
<!-- This function defines the price of each product. To add
the product edit further down the page. -->
<script language=JavaScript>
function calc(productNo)
{
if (productNo==1)
{
document.BuyForm.amount.value = 5.00;
document.BuyForm.desc.value = "Product 1";
}
else if (productNo==2)
{
document.BuyForm.amount.value = 10.00;
document.BuyForm.desc.value = "Product 2";
}
// To add a new product price, copy from here
else if (productNo==3)
{
document.BuyForm.amount.value = 15.00;
document.BuyForm.desc.value = "Product 3";
}
//...to here, and paste directly below.
// You will need to alter the 'productNo' and its price
value.
}
</script>

<h1>One-Stop Shop</h1>

<!-- This table provides layout for the products listed on
the webpage. -->
<table align="center" cellpadding="3" border="2">
<tr>
```

```
<td>Product 1</td>
<td> Price: &pound;5.00</td>
<td><input type="submit" value="Buy button"
onClick="calc(1)"></td></tr>
<tr>
<td>Product 2</td>
<td>Price: &pound;10.00</td>
<td><input type="submit" value="Buy button"
onClick="calc(2)"></td>
</tr>
<!-- To add a new product, copy from here... -->
<tr>
<td>Product 3</td>
<td>Price: &pound;15.00</td>
<td><input type="submit" value="Buy button"
onClick="calc(3)"></td>
</tr>
<!-- ...to here, and paste directly below. You will need to
alter three things: the product number, price, and calc (put
product number here) -->
</table>
</form>
</body>
</html>
```

Test Card Numbers

The following card numbers can be used when you make test transactions in test environments only - do not use them in live, production environments:

<i>card type</i>	<i>card number</i>	<i>number length</i>	<i>issue no length</i>
Mastercard	5100080000000000	16	0
Visa Delta - UK	4406080400000000	16	0
Visa Delta - Non UK	4462030000000000	16	0
Visa	4911830000000	13	0
Visa	4917610000000000	16	0
American Express	3700002000000000	15	0
Diners	36700102000000	14	0
JCB	3528000700000000	16	0
Visa Electron (UK only)	4917300800000000	16	0
Solo	6334580500000000	16	0
Solo	633473060000000000	18	1
Discover Card	6011000400000000	16	0
Laser	630495060000000000	18	0
Maestro	6759649826438453	16	0
Visa Purchasing	4484070000000000	16	0



Note that Visa Purchasing transactions are treated as Visa credit card transactions.

German ELV

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

Account number: 12345678

Bank code: 10000000

Bank name: Bundesbank

Bank residence: Berlin

<i>card type</i>	<i>bank code</i>	<i>account number</i>
ELV	20030000	92441196
ELV	43050001	122108525
ELV	30070024	5929120

Please note that ELV must be activated in the production environment for merchants who would like to test ELV transactions.

Payment Result - to Shopper

Purpose

This section describes the types of payment results sent to a shopper when a payment has been processed.

Once a payment has been processed, a result page is displayed to the shopper and a confirmation email showing the transaction details is sent to the email address they supplied during the purchase. Note that this is not a VAT receipt.

You can replace the default result pages by uploading your own versions, alternatively you can supply your own versions using the Payment Notifications feature. You can also add your own text to the confirmation email.



Note that we cannot guarantee that the shopper will receive the email as we cannot verify the accuracy of the email address supplied, but they should get the result page (provided that their browser remains connected).

Use

Please refer to the following sections for more details about the results sent to the shopper:

- [Shopper's Result Pages - changing what is displayed to the shopper after payment](#)
- [Shopper's Email Receipt - the email we send to the shopper](#)
- [Uploading Result Pages - how to upload the result pages](#)
- [Payment Notifications Result - making use of the Payment Notifications feature](#)

Shopper's Result Pages

When a transaction has been processed, one of two result pages will be returned to the shopper: a successful transaction page or a cancelled transaction page.

By default, we send out a file called `resultY.html` for a successful transaction and `resultC.html` for a cancelled transaction, but there are two ways in which you can supply your own customised response.

- **Uploading** - You can upload your own versions of the `resultY.html` and `resultC.html` files to our server to replace the default result pages. For more information, please refer to [Uploading Result Pages](#).
- **Payment Notifications Result** - If you are using the Payment Notifications feature, you can respond to our Payment Notifications message with a script of your own, by sending an appropriate page of HTML, which will be displayed to the shopper instead of the `resultY.html` or `resultC.html` files. For more information, please refer to [Payment Notifications](#).

Shopper's Email Confirmation

A confirmation email is sent to the address supplied by the shopper during the transaction. Note that this is not a VAT receipt. You can add your own text to this email with the Confirmation Email option in the Payment Pages Editor.



Note that we are legally required to provide some information in the email text, such as your company name.

Uploading Result Pages

You can replace the default `resultY.html` and `resultC.html` files by uploading your own versions to our payment service, via the **File Management** option in the Payment Page Editor - available in the Merchant Interface after selecting the **Installations** option in the left-hand menu.

Creating the Files

We recommend that you create your own versions of these files. Although you can download our default versions of these files to use as the basis of your own files, you should be aware that they include complex tags, accordingly it may be easier to start from scratch.

But, whatever method you use, please ensure that the file names match those we use: the file called `resultY.html` is for a successful transaction and the file called `resultC.html` is for a cancelled transaction.

The result page automatically displays with the header and footer stored for your installation on our payment service. These will either be our defaults or your own, if you have created and uploaded them (via the Payment Page Editor).

We always put a standard banner into the final payment result page. The banner contains legal information that must be conveyed to the shopper. It is inserted at the `WPDISPLAY` tag, as follows:

```
<WPDISPLAY ITEM="banner" >
```

The rest of the page is under your control, as shown in the examples below.

Please also refer to the [Customising Guide \(Advanced\)](#) for more details about adding the banner.

Parameter Substitution

Before being passed to the shopper, we process the result page for parameter substitution. This enables you to embed details of the transaction in the page. The examples below show parameter substitution at the `<WPDISPLAY>` tags.

For more information about substitution, please refer to [Tags Used in Substitution](#).



Note that you can also use custom parameters in your `resultY.html` or `resultC.html` files, for more information please refer to [Custom Parameters](#).

Example - resultC.html : Cancelled Transaction

```
<html>
<head>
<title>Payment Cancelled</title>
</head>
<WPDISPLAY FILE=header.html>
<h1>My Company Ltd.</h1>
<br>Your order has been cancelled<br>
<a href="www.mycompany.com">Click here to return to my
homepage</a>
<WPDISPLAY ITEM=banner>
<WPDISPLAY FILE=footer.html>
</html>
```

Example - resultY.html : Successful Transaction

```
<html>
<head>
<title>Thank you for your payment</title>
</head>
<WPDISPLAY FILE=header.html>
<h1>My Company Ltd.</h1>
<WPDISPLAY ITEM=name>, thank you for your payment
of<WPDISPLAY
ITEM=amountString> for <WPDISPLAY ITEM=desc>. Your goods
will be
shipped to you within three working days
<WPDISPLAY ITEM=banner>
<WPDISPLAY FILE=footer.html>
</html>
```

Payment Result - to You

Purpose

This section describes the types of payment result sent to you when a payment has been processed. These are shown in the table below.



Note that the shopper can cancel at any point in the payment process, including when a card is declined by its issuer. The shopper always has the choice of trying another method of payment, such as another card.

Use

When a payment has been processed, either through to success or to cancellation, information about the processed transaction is made available to you in three ways:

- **Merchant Interface** - You can login to the Merchant Interface to view details of the transaction.
- **Email Confirmations** - We send you an email about each transaction. The confirmation email is sent to the address you specified for the merchant code receiving the payment.
- **Payment Notifications Message** - If you have Payment Notifications (Callbacks) enabled, a message will be sent to you whenever a payment transaction is completed. All the details of the transaction are sent back to you in the message. For more information, please refer to [Payment Notifications \(Callbacks\)](#).

Types of Payment Results

This table shows the types of payment result sent to you when a payment has been processed.



Note that the right-most column in the table is only relevant if you have Payment Notifications enabled, for more information, please refer to [Payment Notifications \(Callbacks\)](#).

Hosted Payment Page service

<i>status/condition</i>	<i>transaction ID generated (view with the MAI)</i>	<i>email sent</i>	<i>Payment Notification sent</i>
successful	Yes	Yes	Yes
declined	Yes	No	No
cancelled	No	No	Yes
browser closed connection broken or similar	No	No	No

Error Messages and Problems

Introduction

The error messages and problems described in this section have been included in response to specific customer queries.

- ⇒ **I Copied the HTML Example Code and It Doesn't Work**
- ⇒ **Merchant Has No Suitable Accounts for This Purchase**
- ⇒ **I Get a Processing Error When I Attempt a Transaction**
- ⇒ **Some of My Chosen Currencies are Missing**

I Copied the HTML Example Code and It Doesn't Work

If you have problems using the example HTML code, please check the following points:

- check that you have copied the code correctly, in particular, take care with spelling the parameters and note that they are case-sensitive: INSTID is **not** the same as instld.
- check that the punctuation and spacing of your code matches the example HTML code

For a full list of parameters and their descriptions, please refer to [eHosted Payment Page \(Redirect Model Parameters\)](#).

Merchant Has No Suitable Accounts for This Purchase

This error message can be generated in response to a premature attempt to submit order details to the Production Environment before your merchant code (account) has been enabled.

Initially, your live merchant code is not enabled for transactions, so there is no risk of accidentally triggering a live transaction until you and we have agreed that you are ready to go live. You can develop and test your site using the test environment, which behaves in the same way as the live Production Environment - except that no money is actually transferred when you submit order details.

If you try to submit order details to the Production Environment before you go live, you will receive this error message. You may also get the error message, "No currency value/variable is being passed across".

For more information about the test environment, please refer to the [Test and Go Live Guide](#).

I Get a Processing Error When I Attempt a Transaction

This error message usually occurs while a merchant is in the process of integrating.

The most common causes are that an unknown character is being included in the values for the parameters, or no value is being supplied at all. For example, consider the following:

```
<input type=hidden name="instId" value="#12345">
```

The # symbol would cause an error to occur when the content of the form is posted to our payment service, because the server is not expecting the value of `instId` to contain a non-numeric character.

Some of My Chosen Currencies are Missing

If some of your chosen currencies are missing from the Currency Selector Box in the payment page, please check that you are using the correct Installation ID for the transaction. If this is correct, then your currencies may not have been added to our system, please contact us.

For more information on currencies, please refer to [Different Currencies](#).

Redirect Model Parameters

Purpose

This section describes the Hosted Payment Page (HTML Redirect) Parameters. They are grouped in terms of their intended use; there are tables of parameters used in the initiating purchase transactions, a table of parameters to be used in the payment pages and a table of parameters for use with the results pages.



Note that each parameter must be named precisely as shown. Also note that the parameters are case-sensitive.

All parameters are available for substitution. For more information, please refer to [Tags Used in Substitution](#).

For information about Recurring Payments (FuturePay) parameters, please refer to the [Recurring Payments Service \(FuturePay\) Guide](#).

Use

Please refer to the following sections for more details about RBS WorldPay parameters:

- ⇒ [**Mandatory Parameters**](#) - the parameters you must include in an order details submission
- ⇒ [**Initiating Purchase Transaction Parameters**](#) - optional parameters for the order details submission
- ⇒ [**Payment Page Parameters**](#) - parameters you can use in the payment pages
- ⇒ [**Payment Result Parameters**](#) - parameters you can use in the result pages the shopper sees
- ⇒ [**FuturePay Parameters**](#) - you can use FuturePay parameters in the order details submission
- ⇒ [**Custom Parameters**](#) - what they are and how to use them

Mandatory Parameters

Although there are parameters that are optional with regard to order details submissions, some of them are mandatory with regard to the payment pages. For instance, the shopper's address can be optionally included in the order details, but if it is not then it **must** be provided by the shopper in the payment pages.

Thus, there are two senses of mandatory parameter:

1. where they are mandatory with regard to the **order details submission**,
2. where they are mandatory with regard to the **payment pages**.

Initiating Purchase Transaction Parameters

The parameters described in these tables are intended to be used when submitting order details; doing so initiates a purchase transaction. Note that some parameters are mandatory.

Where a length in characters is given, this is the maximum length that the parameter can be - anything longer will be truncated to this length.

Mandatory Order Details Parameters

The following four parameters **must** be included in the order details you submit.

<i>parameter name</i>	<i>type</i>	<i>description</i>
instld	integer	Your Installation Id.
cartld	255 char	Your own reference number for this purchase. It is returned to you along with the authorisation results by whatever method you have chosen for being informed (email and / or Payment Notifications).
amount	Decimal	A decimal number giving the cost of the purchase in terms of the major currency unit e.g. 12.56 would mean 12 pounds and 56 pence if the currency were GBP (Pounds Sterling). Note that the decimal separator must be a dot (.), regardless of the typical language convention for the chosen currency. The decimal separator does not need to be included if the amount is an integral multiple of the major currency unit. Do not include other separators, for example between thousands.
currency	3 char	3 letter ISO code for the currency of this payment - please refer to the appendix Currency Codes .

Optional Order Details Parameters

The following parameters are optional with regard to the order details submission. Note that some are mandatory with regard to the **payment pages** - if they are not supplied in the order details then they must be entered in the payment pages by the shopper.



Also note that at your request we can assign mandatory status to the `postcode` parameter to help prevent fraud. When this parameter is set to mandatory, a shopper has to supply postcode details and a card security code in the payment pages.

Redirect Model Parameters

<i>parameter name</i>	<i>type</i>	<i>mandatory in payment page?</i>	<i>description</i>
address	255 char	Yes	The shopper's address. Encode newlines as "
" (the HTML entity for ASCII 10, the new line character). If this is not supplied in the order details then it must be entered in the payment pages by the shopper.
country	2 char	Yes	The shopper's country, as 2 character ISO code, uppercase. Please refer to the appendix Country Codes . If this is not supplied in the order details then it must be entered in the payment pages by the shopper.
desc	255 char	-	A textual description of this purchase, up to 255 characters. This is used in web-pages, statements and emails for yourself and the shopper.
resultFile	string	-	The name of one of your uploaded files, which will be used to format the result. Please refer to Configuring Your Installation . If this is not specified, resultY.html or resultC.html are used as described in Payment Result - to You .
acclId<n>	string	-	This specifies which merchant code should receive funds for this payment. By default our server tries acclId1.
authMode	char	-	This specifies the authorisation mode to use. This is only needed if you have merchant codes with different authorisation modes, in order to specify which type of merchant code to use. If there is no merchant code with a matching <code>authMode</code> then the transaction is rejected. The values are "A" for a full auth, or "E" for a pre-auth. In the payment result this parameter can also take the value "O" when performing a post-auth.

<p>testMode</p>	<p>integer</p>	<p>-</p>	<p>A value greater than 0 specifies that this is a test payment. Specify the test result you want by entering REFUSED, AUTHORISED, ERROR, or CAPTURED in the <code>name</code> parameter.</p> <p>When you submit order details using the <code>testMode</code> parameter and the URL for the live Production Environment you will be presented with a page asking you if you want to redirect the order details to the Test Environment - select the Redirect button if you do.</p> <p>If you submit the order details to the live, production environment our systems will attempt to debit merchant codes (accounts).</p> <p>Reversing transactions such as these, and adjusting accounts, will cause unnecessary work for us as well as yourself.</p> <p>Set this parameter to 0 (zero) or omit it for a live transaction.</p> <p>Please also refer to the Test and Go Live Guide.</p>
<p>authValidFrom</p>	<p>integer</p>	<p>-</p>	<p>This specifies a time window within which the purchase must (or must not) be completed, eg. if the purchase is a time-limited special offer. Each of these parameters is a time in milliseconds since 1 January 1970 GMT - a Java long date value (as from <code>System.currentTimeMillis()</code> or <code>Date.getTime()</code>), or <code>1000 * a C time_t</code>. If <code>from < to</code>, then the authorisation must complete between those two times. If <code>to < from</code>, then the authorisation must complete either before the <code>to</code> time or after the <code>from</code> time. Either may be zero or omitted to give the effect of a simple "not before" or "not after" constraint. If both are zero or omitted, there are no restrictions on how long a shopper can spend making their purchase (although our server will time-out their session if it is idle for too long).</p>
<p>authValidTo</p>	<p>integer</p>		

Redirect Model Parameters

name	40 char	-	<p>The shopper's full name, including any title, personal name and family name.</p> <p>Note that if you do not pass through a name, and use Payment Notifications (Callbacks), the name that the cardholder enters on the payment page will be returned to you as the value of name in the Payment Notifications message.</p> <p>Also note that if you are sending a test submission you can specify the type of response you want from our system by entering REFUSED, AUTHORISED, ERROR or CAPTURED as the value in the name parameter. You can also generate an AUTHORISED response by using a real name, such as, J. Bloggs. For more information please refer to the Test and Go Live Guide.</p>
postcode	12 char	Can be set to mandatory	<p>The shopper's postcode.</p> <p>Note that at your request we can assign mandatory status to this parameter. That is, if it is not supplied in the order details then the shopper must enter it in the payment pages.</p>
tel	30 char	-	The shopper's telephone number.
fax	30 char	-	The shopper's fax number.
email	80 char	-	The shopper's email address.

Display Parameters

The following parameters control the appearance of the payment pages.

<i>parameter name</i>	<i>type</i>	<i>description</i>
fixContact	needs no value	If present, this causes contact details to be displayed in non-editable format. You must ensure that all mandatory contact details are submitted in your initial request.
hideContact	needs no value	If present, this causes contact details to be hidden. You must ensure that all mandatory contact details are submitted in your initial request. Existing merchants should set the following message files to empty strings for the feature to work: cont.instr.existing, cont.instr.new, cont.heading
hideCurrency	needs no value	If present, this causes the currency drop down to be hidden, so fixing the currency that the shopper must purchase in.
lang	6 char	The shopper's language choice, as a 2-character ISO 639 code, with optional regionalisation using 2-character country code separated by hyphen. For example "en-GB" specifies UK English. The shopper can always choose a language on our pages or via browser preferences but if your site has already made this choice then you can make things more convenient by submitting it to us.
noLanguageMenu	needs no value	This suppresses the display of the language menu if you have a choice of languages enabled for your installation but want the choice to be defined by the value of the <code>lang</code> parameter that you submit. Please contact your local Technical Support department if you would like this facility enabled on your account.
withDelivery	needs no value	Displays input fields for delivery address and mandate that they be filled in.

The `subst` Parameter

This parameter is intended for use during testing. It is only relevant if you are creating your own messages files.

<i>parameter name</i>	<i>type</i>	<i>description</i>
subst	string: "yes" or "no"	If the value is "no" then message substitution is turned off. This means that you see the names of the message properties from the <code>messages_xx.properties</code> file used to create the page. This situation persists until you submit a payment with <code>subst=yes</code> or your session is ended.

Payment Page Parameters

All of the parameters shown in the [Initiating Purchase Transaction Parameters](#) tables are available for use in the payment pages. The following parameters are also available for use in the payment pages.

Please also refer to the [Display Parameters](#) table in the section [Initiating Purchase Transaction Parameters](#).

<i>parameter name</i>	<i>type</i>	<i>description</i>
amountString	Variable length char string	An HTML string produced from the amount and currency that were submitted to initiate this purchase.
countryString	Variable length char string	The full name of the country, derived from the country code submitted to initiate this purchase or supplied by the shopper.
compName	Variable length char string	Name of the company associated with this installation.

Payment Result Parameters

All of the parameters shown in the [Initiating Purchase Transaction Parameters](#) tables, and the [Payment Page Parameters](#) table are available for use in the payment results. The following parameters are also available for use in the payment results.

At the payment result stage of processing, the contact details for the shopper will be those that were used for authorising the payment.

The parameters `authAmount`, `authCurrency`, etc., are set to those of the actual transaction carried out. Note that these could be different to the values submitted to us when the transaction was initiated, as the shopper may have chosen a different currency when in the payment pages.

Where a length in characters is given, this is the maximum length that the parameter can be - anything longer will be truncated to this length.

<i>parameter Name</i>	<i>type</i>	<i>description</i>
transId	integer - 16 digits	The RBS WorldPay ID for this transaction.
futurePayId	integer - 16 digits	The RBS WorldPay ID for a FuturePay agreement (where relevant).
transStatus	1 char	Result of this transaction - "Y" for a successful payment authorisation, "C" for a cancelled payment (note that as a merchant you will never see a declined payment, as the shopper is always given the option of retrying with another means of payment, or else cancelling the payment).
transTime	long integer	The time of this transaction in milliseconds since the start of 1970 GMT. This is the standard system date in Java, and is also 1000x the standard C <code>time_t</code> time.
authAmount	decimal	The amount that this transaction was authorised for, in the currency given as <code>authCurrency</code> .
authCurrency	3 char	The currency used for authorisation.
authAmountString	Variable length char string	HTML string produced from authorisation amount and currency

rawAuthMessage	string	The text received from the bank (typically including an authorisation code, or a reason for failure).										
rawAuthCode	1 char	A single-character bank (or internal RBS WorldPay) authorisation code. This is retained for backward compatibility. 'A' means 'authorised' and is directly equivalent to <code>transStatus='Y'</code> . Failed transactions may have a variety of auth codes which are usually explained more fully in the <code>rawAuthMessage</code> parameter.										
callbackPW	string	The Payment Notifications password, if you have set it in our database via the Merchant Interface. This is only available in the parameters sent in the Payment Notifications message. It is not available for substitution into the page sent to the shopper.										
cardType	string	The type of card used by the shopper.										
countryMatch	1 char	<p>A single character describing the result of the comparison of the shopper's contact country (where supplied) and the issue country of the card used by the shopper (where available). Note that this parameter is retained for backward compatibility - equivalent information is now provided as part of the AVS results (see AVS below).</p> <p>Key:</p> <table border="1" data-bbox="774 1220 1343 1563"> <tr> <td>Y</td> <td>Match</td> </tr> <tr> <td>N</td> <td>No match (that is, mismatch)</td> </tr> <tr> <td>B</td> <td>Comparison not available</td> </tr> <tr> <td>I</td> <td>Contact country not supplied</td> </tr> <tr> <td>S</td> <td>Card issue country not available</td> </tr> </table>	Y	Match	N	No match (that is, mismatch)	B	Comparison not available	I	Contact country not supplied	S	Card issue country not available
Y	Match											
N	No match (that is, mismatch)											
B	Comparison not available											
I	Contact country not supplied											
S	Card issue country not available											
AVS	4 char	<p>The Address Verification System (AVS) provides the ability to check the shopper's billing address and postcode against the card issuer's details. The AVS results combine the results of the AVS check with the results of a CVV check.</p> <p>CVV is a three-digit number printed at the end of the signature strip on the vast majority of credit/debit cards. The CVV is a unique number that cannot be replicated by fraudsters, and for this reason, it is commonly known as the 'security code'.</p>										

		<p>The results of Security Code and Address Verification are provided as a four character string in which each character reports the result of a particular verification/check.</p> <p>The characters give the results of the following checks:</p> <table border="1" style="margin-left: 40px;"> <tr> <td style="text-align: center;">1st character</td> <td>Card Verification Value check</td> </tr> <tr> <td style="text-align: center;">2nd character</td> <td>Postcode AVS check</td> </tr> <tr> <td style="text-align: center;">3rd character</td> <td>Address AVS check</td> </tr> <tr> <td style="text-align: center;">4th character</td> <td>Country comparison check (please also refer to countryMatch)</td> </tr> </table> <p>The possible values for each result character are:</p> <table border="1" style="margin-left: 40px;"> <tr> <td style="text-align: center;">0</td> <td>Not supported</td> </tr> <tr> <td style="text-align: center;">1</td> <td>Not checked</td> </tr> <tr> <td style="text-align: center;">2</td> <td>Matched</td> </tr> <tr> <td style="text-align: center;">4</td> <td>Not matched</td> </tr> </table>	1st character	Card Verification Value check	2nd character	Postcode AVS check	3rd character	Address AVS check	4th character	Country comparison check (please also refer to countryMatch)	0	Not supported	1	Not checked	2	Matched	4	Not matched
1st character	Card Verification Value check																	
2nd character	Postcode AVS check																	
3rd character	Address AVS check																	
4th character	Country comparison check (please also refer to countryMatch)																	
0	Not supported																	
1	Not checked																	
2	Matched																	
4	Not matched																	

Recurring Payments (FuturePay) Parameters

You can include Recurring Payments (also known as FuturePay) parameters in an order details submission. For instance, to set-up a Recurring Payments agreement, where a number of payments are to be made at intervals, such as in a subscription. To do this, specify the Recurring Payments parameters within the order details, as shown in the example below.

```
<!-- Futurepay parameters start here -->
<INPUT TYPE=HIDDEN NAME=futurePayType VALUE="regular">
<INPUT TYPE=HIDDEN NAME=option VALUE=1>
<INPUT TYPE=HIDDEN NAME=startDelayMult VALUE=3>
<INPUT TYPE=HIDDEN NAME=startDelayUnit VALUE=2>
<INPUT TYPE=HIDDEN NAME=noOfPayments VALUE=11>
<INPUT TYPE=HIDDEN NAME=intervalMult VALUE=1>
<INPUT TYPE=HIDDEN NAME=intervalUnit VALUE=3>
<INPUT TYPE=HIDDEN NAME=normalAmount VALUE=17.99>
<INPUT TYPE=HIDDEN NAME=initialAmount VALUE=10.00>
```

For more information, please refer to Recurring Payments sections in the Payment Notifications Guide. For information about FuturePay parameters, please refer to the different agreement types in the [Recurring Payments Service \(FuturePay\) Guide](#).

Custom Parameters

You can create your own custom parameters, which will be passed through our payment service and returned to your server in a Payment Notifications (Callback) message. For more information, please refer to the [Payment Notifications Guide](#).

Custom parameters can also be used in your own versions of the `resultY.html` or `resultC.html` pages that are returned to the shopper in response to their payment.

The custom parameters that you want to return to yourself in a Payment Notifications message should have names starting with "M_", and custom parameters to be used just in the shopper's result page should start with "C_". To use a parameter for both, use a prefix of "MC_" or "CM_".

- Custom "C_" parameters are available for use in the result pages, but are NOT available to your Payment Notifications script.
- Custom "M_" parameters are available to your Payment Notifications script, but are NOT available for use in the result pages.
- Custom "MC_" and "CM_" parameters are available to both your Payment Notifications script and the result pages.



Note that custom "M_" or "MC_" parameters that you submit in order details will be returned in the Payment Notifications message. Also note that "C_" parameters are not returned in the Payment Notifications message, but are displayed to the shopper in the result page.

Custom Parameter Examples

```
<input type=hidden name="M_my-custom-var" value="sent in the HTML  
Payment Notifications message">  
<input type=hidden name="C_my-custom-var" value="displayed  
on the Shopper's page">  
<input type=hidden name="MC_my-custom-var" value="M_ and MC_  
combined">
```

Tags Used in Substitution

Purpose

This section describes the substitutions you can make in the files used in the construction of pages or emails.

All parameters are available for substitution, provided that the transaction has reached the appropriate stage. For instance, there would not be a `transStatus` value available before a transaction had been processed to completion. If you use an unavailable parameter in substitution then the default as specified in the substitution tag will be used.

For details of the parameters available, please refer to [Hosted Payment Page \(HTML Redirect\) Parameters](#).

Use

Please refer to the following sections for more details about substitution tags:

- ⇒ [Parameter and File Substitution](#) - you can substitute parameter values and embed files
- ⇒ [Substituting the Value of Parameters](#) - how to substitute parameter values
- ⇒ [Substituting the Value of a Display Property](#) - how to substitute display parameter values
- ⇒ [Embedding the Contents of a File](#) - embed a file
- ⇒ [Embedding Tags Within Tags](#) - nesting tags
- ⇒ [Custom Parameter Substitution](#) - making use of your own parameters

Parameter and File Substitution

All files that are used in the construction of pages or emails are subject to parameter and file substitution. Substitution is carried out using special `WPDISPLAY` tags. These tags can be placed anywhere, including within HTML tags.

Parameters provide information about the transaction. For instance, the parameter `amountString` is the cost of the item or product in the payment pages, and `desc` is the description of the item or product. Other parameters hold other information, such as the shopper's billing address details and the transaction status. Any of these parameters can be substituted using `WPDISPLAY` tags.

For a full list of the available parameters, please refer to [Hosted Payment Page \(HTML Redirect\) Parameters](#).

Substituting the Value of Parameters

In the following example of a `resultY.html` response page, each `ITEM` tag will be replaced with the values of the named parameters:

```
<html>
<form
action="https://secure.wp3.rbsworldpay.com/wcc/purchase"
method=POST>
<head>
<title>Thank you for your payment</title>
</head>
<WPDISPLAY FILE=header.html DEFAULT="<body
bgcolor=#ffffff">">
<h1>My Company Ltd.</h1>
<WPDISPLAY ITEM=name>, thank you for your payment of
<WPDISPLAY ITEM=amountString> for
<WPDISPLAY ITEM=desc>. Your goods will be shipped to you
within three working days.
<WPDISPLAY ITEM=banner>
<WPDISPLAY FILE=footer.html DEFAULT="</body">">
</html>
```

You can also embed your standard header and footer, as well as any graphics you wish. Note that these graphics must be uploaded to our payment service, even if the page is returned as a result page by your server. For more information about uploading graphic files, please refer to the File Management topic in [Configuring Your Installation](#).



Note that if you have used custom parameters in the order details, any "M_" or "MC_" parameters that you submitted will also be returned. Also note that "C_" parameters are not returned in the Payment Notifications message, but are displayed to the shopper in the result page. For more information, please refer to [Custom Parameters](#).

Substituting the Value of a Display Property

The following HTML will construct a `font` tag containing the value of the parameter named `title.bg` from `display.properties`. This means that you can use the same parameters that are used in the payment pages in your own templates. You can also add parameters of your own to `display.properties` and use these in your templates.

```
<WPDISPLAY DISP="title.bg" DEFAULT="<font color=#000000">
PRE="<font color=" POST=">">
```

Embedding the Contents of a File

In the example below, the FILE tag will be replaced with the contents of the named file `inner.txt`, which should be in your upload area. The embedded file can contain any WorldPay tags except for another file embedding command.

```
<WPDISPLAY FILE="inner.txt" DEFAULT="not found">
```

Embedding Tags Within Tags

The strings given as the values of WPDISPLAY attributes can themselves contain WPDISPLAY tags - except for file embedding tags. For example, given a `cartId` of 123, the line:

```
<WPDISPLAY FILE="blurb<WPDISPLAY ITEM=cartId>.txt">
```

will expand to:

```
<WPDISPLAY FILE="blurb123.txt">
```

which then results in the embedding of the file `blurb123.txt`.

Do not use the same type of quotes in the inner tag: either omit them altogether as in the example above (only viable for single words) or use the other kind (single versus double quote), as shown below.

```
<WPDISPLAY FILE="blurb<WPDISPLAY ITEM='cartId'>.txt">
```

Custom Parameter Substitution

Any "C_" or "MC_" custom parameters that you supplied in the initial submission to us are also available for substitution.

For example, assume that you have assigned a custom parameter `MC_my-custom-var` with the value "Don't forget there is 10% off all gift packs on Friday 10th Sept!". Then, in the following HTML line:

```
<WPDISPLAY ITEM="MC_my-custom-var" DEFAULT=" " PRE="<b>"  
POST="</b>">
```

the PRE bold tag and POST bold-off tag will bracket the ITEM tag, so that the substitution will be done as follows:

```
<b>Don't forget there is 10% off all gift packs on Friday  
10th Sept!</b>
```

which will display as: **Don't forget there is 10% off all gift packs on Friday 10th Sept!**

Enhancing Security

Purpose

This section describes the MD5 message-digest algorithm and its implementation for both ready-built shopping carts and bespoke stores.

Use

Please refer to the following sections for more details about MD5:

- ⇒ [What is MD5? - a brief overview](#)
- ⇒ [Implementation and Prerequisites - how to implement for off-the-shelf or bespoke shopping carts](#)
- ⇒ [Additional Security - going the extra yard](#)

What is MD5?

As part of our ongoing commitment to safe online trading, we have introduced a new security feature, MD5, for merchants using the Hosted Payment Page (HTML Redirect) service.

This new feature enables merchants to add an encrypted signature to each order details submission they send to us. This protects the order details from unauthorised tampering.

MD5 is an algorithm designed to create digital signatures that can be used to verify data integrity. The signatures are created by taking a string or message of arbitrary length and producing an encrypted output as a 128-bit "fingerprint" or "Message Digest".

The signatures are highly individual. The designers of MD5 conjecture that it is "computationally infeasible" that two inputs could produce the same output or that a message could be decrypted from the signature alone. It is ideal for securing the transfer of purchase information.

Using MD5 with Your Integration

Using MD5, you can choose which order details parameters you particularly want us to check for unauthorised tampering.

The values of these parameters are encrypted along with a "secret" key that is only known to yourself and us. This encrypted signature is sent as an additional parameter in your order details submission along with another parameter telling us which purchase details have been encrypted and in which order.

Hosted Payment Page service

When we receive the order details, the MD5 value is checked to ensure that it still corresponds with the transaction parameter values submitted. If there are any discrepancies, the transaction is rejected.



Note that you must still include all mandatory parameters in the order details submission.

Implementation and Prerequisites

To incorporate MD5 security in your store, your store must either:

- use an off-the-shelf shopping cart that has been updated to use MD5 (please consult your cart vendor for advice)
or,
- be a bespoke store that has been generated using a programming language that has an MD5 library available, such as C, C++, Visual Basic, PHP, ASP and Perl **but not** HTML or Javascript.



Note that HTML and Javascript are unsuitable for this feature because it is essential the encryption process happens server-side to ensure the security of the data.

Implementation for Shopping Carts

To add MD5 encryption to your site if you have used an off-the-shelf shopping cart, you will need to do the following:

1. **Enable MD5 functionality in your store as advised by your shopping cart provider.** Implementation will vary according to the package that you are using, but you are at least likely to need to select which fields are included in the encrypted signature, you will also have to supply a "secret" to be used in encrypting the signature. This should be a string (spaces are permitted) of up to 16 characters, known only to yourself and to us.
2. **Specify your secret to us.** This secret must be entered into the **MD5 secret for transactions** field in the Integration Setup for your installation - via the Merchant Interface.

Note that if you have specified a secret, you **must** sign all transactions correctly otherwise they will be rejected. If you wish to disable the MD5 functionality at any point, please contact us.

Implementation for Bespoke Stores

To add MD5 encryption to your site if you have built a bespoke store, you will need to do the following:

1. **Choose a "secret" to be used in encrypting and decrypting the signature.** This should be a string (spaces are permitted) of up to 16 characters, known only to yourself and to us. The string should be static rather than dynamically generated. This secret must be sent to our technical support team and you will need to enter it in the **MD5 secret for transactions** field in the Integration Setup for your installation - via the Merchant Interface.

Note that if you have specified a secret, you **must** sign all transactions correctly otherwise they will be rejected. If you wish to disable the MD5 functionality at any point, please contact us.

2. **Choose which of the order details parameters you wish to protect with the signature.** You can encrypt whichever parameters you like but, if it is dynamically generated, we particularly recommend that you include: `amount`, `currency`, `instId` and the `cartId` parameter.
3. **Add a new parameter named `signatureFields` to the order details.** This parameter should list the fields you want to include in the encrypted signature as a colon separated list.

```
<input type=hidden name=signatureFields
value="amount:currency:cartId">
```

4. **Construct a string consisting of the secret and the values of the parameters you want to encrypt.** The items in this string should be separated by colons. For example, if you choose the word "champion" as your secret and the fields you have chosen to encrypt have the following values: `amount=123.00`, `currency=GBP` and `cart Id=ABC123`, your string should look like this:

```
champion:123.00:GBP:ABC123
```

It is **vital** that you list the parameter values in the same order in which you listed them in the `signatureFields` parameter.

5. **Calculate an MD5 signature from this string and include the signature in the order details.** The MD5 algorithm will return a hexadecimal value 32 characters long. This digital signature should be included in the order details submission as the value of a parameter named `signature`.

```
<input type=hidden name=signature
value="3365b7d3c5ac46df8a0f359f65e16317">
```



Note that the way you use MD5 encryption is dependent on the code you have used to build your store. We regret we are not able to assist you in incorporating the code into your store but only in how to send that information to us. For more information about MD5, please refer to [MD5 Reference Sites](#).

Additional Security

As an additional security feature in the Hosted Payment Page (HTML Redirect) you can specify a time period for which the purchase transaction is valid. Using this functionality in conjunction with MD5 increases security by limiting the amount of time available for an unauthorised user to attempt to decrypt the transaction.

To specify the time after which the transaction will be deemed invalid, use the parameter `authValidTo`. The value of `authValidTo` should be a date and time given in the Unix time standard: milliseconds since 1st January 1970. For example,

```
<input type=hidden name=authValidTo value="938736000000">
```

Fifteen minutes is a commonly-used period of validity but you might choose to reduce this time period to 10 or even 5 minutes.

To implement this feature you should:

1. ensure the clock on your server is synchronised to an accurate time source, for example, by using NTP, at the URL <http://www.eecis.udel.edu/~mills/ntp/servers.html>
2. take a time reading from your server at the time of the transaction and convert it to the Unix time standard, if necessary
3. multiply by 1000 the number of seconds you want the order details to be valid and add this to the time. This will be the value of `authValidTo`.

MD5 Reference Sites

If you would like to find out more about MD5 and how to incorporate it into your store, please refer to the following web sites:

<http://www.rsasecurity.com/rsalabs/faq/3-6-6.html>: RSA Laboratories definition of MD5,

<http://userpages.umbc.edu/~mabzug1/cs/md5/md5.html>: details on how to implement MD5 in many program languages,

<http://www.fags.org/rfc/rfc1321.html>: specification of the MD5 function provided by the inventor at MIT.

Appendices

Introduction

The appendices available for this guide are listed below.

- ⇒ **Exchange Rates** - describes how to get the latest exchange rates
- ⇒ **Currency Code** - lists the ISO currency codes that should be used to describe the currencies used by your shop
- ⇒ **Country Code** - lists the ISO country codes that should be used to describe the shopper's country

Introduction

The appendices available for this guide are listed below.

- ⇒ **Exchange Rates** - describes how to get the latest exchange rates
- ⇒ **Currency Code** - lists the ISO currency codes that should be used to describe the currencies used by your shop
- ⇒ **Country Code** - lists the ISO country codes that should be used to describe the shopper's country

Exchange Rates

The exchange rates we use are updated every 24 hours at midnight GMT. You can obtain a file containing the exchange rates that are valid for the current day.

This section includes information on the following topics:

- ⇒ requesting the latest exchange rates file
- ⇒ the exchange rates file.

Request the Latest Exchange Rates File

To obtain the latest exchange rates file, using an Installation ID of "1234", you would need to request the following URL:

<https://secure.wp3.rbsworldpay.com/wcc/info?op=rates&instId=1234>

This will return the latest available exchange rates. These are usually for the current day but if any data is missing then data from previous days is used and the returned field "allRatesCurrent" is set to false, "allRatesCurrent" is true if all the data is for the current day.

Hosted Payment Page service

Note that if you have specified an **Info servlet password** with the Merchant Interface you will need to include it as an `infoPW` parameter in the URL. For example, if your Info servlet password is "abcd", you would need to request the following URL:

```
https://secure.wp3.rbsworldpay.com/wcc/info?op=rates&instId=1234&infoPW=abcd
```

The Exchange Rates File

If an exchange rates file is successfully generated then it is returned as a http response with content-type "text/plain" and has the following format:

```
#Exchange rates for installation 1234
#Fri Sep 28 10:25:37 GMT+00:00 2001
rateDateMillis=1001635200000
GBP_SEK=16.125164729479625
GBP_GBP=1.0
allRatesCurrent=true
GBP_JPY=174.28009178686133
rateDateString=2001-09-28
GBP_NZD=3.700443299768331
GBP_EUR=1.6173201897996292
GBP_CHF=2.3733538756347174
GBP_AUD=3.0421914811575412
GBP_SGD=2.61003882026893
GBP_NOK=12.885850286581093
GBP_CAD=2.328469752766523
GBP_RUR=43.66890329071026
GBP_ZAR=13.06138438810193
GBP_USD=1.4821286404310512
GBP_DKK=12.04706314879048
GBP_HKD=11.560157417867403
```

This is a standard Java properties file format. The lines prefixed by # are comments and are always present at the top of the file. The date in the comment at the head of the file is the date and time of generation of this file. All other lines are of the form: `<property name> = <value>`

The property names for the exchange rates are given as: `<source ISO code>_<destination ISO code>`

To convert from `<destination ISO code>` to `<source ISO code>` you must divide by the given number. Conversely, to convert from `<source ISO code>` to `<destination ISO code>` you must multiply by the given number.

For example, to convert an amount from US Dollars to Pounds Sterling, where the rate is given as GBP_USD=1.4821286404310512, you must divide by the given number 1.4821286404310512.

Rates are given for converting from all your settlement currencies to all your acquisition currencies.



Note that the <property name>=<value> lines are generated in a random order so you cannot rely on any particular ordering of the items.

The other properties present in the file are as follows:

<i>property</i>	<i>definition</i>
rateDateMillis	The date and time from which the rates are valid (always midnight GMT), expressed as milliseconds since the start of 1970. This is the standard system date in Java.
rateDateString	Expresses this date as "yyyy-mm-dd". The rates are valid for 24 hours (that is, until midnight GMT the next day).
allRatesCurrent	Set to true if the rates are all for the date requested. If some rates are from a previous day then it is set to false. This only happens if you request op=rates or op=rates-latest.

If a rates file cannot be generated then the content-type returned is text/html and starts with the string "Error:". Hence, you can check for an error either by testing the content type or by parsing the first line of the content. The rest of the content for an error is a human-readable description of the problem.

ISO Currency Codes

Currencies accepted by our payment service are listed below.

A currency code value is an upper-case three-letter 'ISO 4217' standard currency code.

The full ISO 4217 list can be found at: <http://www.id3.org/iso4217.html>

ISO 4217 Currency Codes

<i>code</i>	<i>name</i>	<i>exponent</i>
ARS	Nuevo Argentine Peso	2
AUD	Australian Dollar	2
BRL	Brazilian Real	2
CAD	Canadian Dollar	2
CHF	Swiss Franc	2
CLP	Chilean Peso	2
CNY	Yuan Renminbi	2
COP	Colombian Peso	2
CZK	Czech Koruna	2
DKK	Danish Krone	2
EUR	Euro	2
GBP	Pound Sterling	2
HKD	Hong Kong Dollar	2
HUF	Hungarian Forint	2
IDR	Indonesian Rupiah	0
JPY	Japanese Yen	2
KES	Kenyan Shilling	2
KRW	South-Korean Won	2
MXP	Mexican Peso	2
MYR	Malaysian Ringgit	2
NOK	Norwegian Krone	2
NZD	New Zealand Dollar	2
PHP	Philippine Peso	2
PLN	New Polish Zloty	2
PTE	Portugese Escudo	2
SEK	Swedish Krone	2
SGD	Singapore Dollar	2
SKK	Slovak Koruna	2

THB	Thai Baht	2
TWD	New Taiwan Dollar	2
USD	US Dollars	2
VND	Vietnamese New Dong	2
ZAR	South African Rand	2

ISO Country Codes

The `country` parameter value is an upper-case two-letter 'ISO 3166' standard country code.

ISO source reference: <http://www.iso.org/iso/en/prods-services/iso3166ma/02iso-3166-code-lists/list-en1.html>

ISO 3166 Two-Letter Country Codes

<i>country name</i>	<i>country parameter value</i>
AFGHANISTAN	AF
ÅLAND ISLANDS	AX
ALBANIA	AL
ALGERIA	DZ
AMERICAN SAMOA	AS
ANDORRA	AD
ANGOLA	AO
ANGUILLA	AI
ANTARCTICA	AQ
ANTIGUA AND BARBUDA	AG
ARGENTINA	AR
ARMENIA	AM

ARUBA	AW
AUSTRALIA	AU
AUSTRIA	AT
AZERBAIJAN	AZ
BAHAMAS	BS
BAHRAIN	BH
BANGLADESH	BD
BARBADOS	BB
BELARUS	BY
BELGIUM	BE
BELIZE	BZ
BENIN	BJ
BERMUDA	BM
BHUTAN	BT
BOLIVIA	BO
BOSNIA AND HERZEGOVINA	BA
BOTSWANA	BW
BOUVET ISLAND	BV
BRAZIL	BR
BRITISH INDIAN OCEAN TERRITORY	IO
BRUNEI DARUSSALAM	BN
BULGARIA	BG
BURKINA FASO	BF

BURUNDI	BI
CAMBODIA	KH
CAMEROON	CM
CANADA	CA
CAPE VERDE	CV
CAYMAN ISLANDS	KY
CENTRAL AFRICAN REPUBLIC	CF
CHAD	TD
CHILE	CL
CHINA	CN
CHRISTMAS ISLAND	CX
COCOS (KEELING) ISLANDS	CC
COLOMBIA	CO
COMOROS	KM
CONGO	CG
CONGO, THE DEMOCRATIC REPUBLIC OF THE	CD
COOK ISLANDS	CK
COSTA RICA	CR
CÔTE D'IVOIRE	CI
CROATIA	HR
CUBA	CU
CYPRUS	CY
CZECH REPUBLIC	CZ

DENMARK	DK
DJIBOUTI	DJ
DOMINICA	DM
DOMINICAN REPUBLIC	DO
ECUADOR	EC
EGYPT	EG
EL SALVADOR	SV
EQUATORIAL GUINEA	GQ
ERITREA	ER
ESTONIA	EE
ETHIOPIA	ET
FALKLAND ISLANDS (MALVINAS)	FK
FAROE ISLANDS	FO
FIJI	FJ
FINLAND	FI
FRANCE	FR
FRENCH GUIANA	GF
FRENCH POLYNESIA	PF
FRENCH SOUTHERN TERRITORIES	TF
GABON	GA
GAMBIA	GM
GEORGIA	GE
GERMANY	DE

GHANA	GH
GIBRALTAR	GI
GREECE	GR
GREENLAND	GL
GRENADA	GD
GUADELOUPE	GP
GUAM	GU
GUATEMALA	GT
GUINEA	GN
GUINEA-BISSAU	GW
GUYANA	GY
HAITI	HT
HEARD ISLAND AND MCDONALD ISLANDS	HM
HOLY SEE (VATICAN CITY STATE)	VA
HONDURAS	HN
HONG KONG	HK
HUNGARY	HU
ICELAND	IS
INDIA	IN
INDONESIA	ID
IRAN, ISLAMIC REPUBLIC OF	IR
IRAQ	IQ
IRELAND	IE

ISRAEL	IL
ITALY	IT
JAMAICA	JM
JAPAN	JP
JORDAN	JO
KAZAKHSTAN	KZ
KENYA	KE
KIRIBATI	KI
KOREA, DEMOCRATIC PEOPLE'S REPUBLIC OF	KP
KOREA, REPUBLIC OF	KR
KUWAIT	KW
KYRGYZSTAN	KG
LAO PEOPLE'S DEMOCRATIC REPUBLIC	LA
LATVIA	LV
LEBANON	LB
LESOTHO	LS
LIBERIA	LR
LIBYAN ARAB JAMAHIRIYA	LY
LIECHTENSTEIN	LI
LITHUANIA	LT
LUXEMBOURG	LU
MACAO	MO
MACEDONIA, THE FORMER YUGOSLAV REPUBLIC	MK

OF	
MADAGASCAR	MG
MALAWI	MW
MALAYSIA	MY
MALDIVES	MV
MALI	ML
MALTA	MT
MARSHALL ISLANDS	MH
MARTINIQUE	MQ
MAURITANIA	MR
MAURITIUS	MU
MAYOTTE	YT
MEXICO	MX
MICRONESIA, FEDERATED STATES OF	FM
MOLDOVA, REPUBLIC OF	MD
MONACO	MC
MONGOLIA	MN
MONTSERRAT	MS
MOROCCO	MA
MOZAMBIQUE	MZ
MYANMAR	MM
NAMIBIA	NA
NAURU	NR

NEPAL	NP
NETHERLANDS	NL
NETHERLANDS ANTILLES	AN
NEW CALEDONIA	NC
NEW ZEALAND	NZ
NICARAGUA	NI
NIGER	NE
NIGERIA	NG
NIUE	NU
NORFOLK ISLAND	NF
NORTHERN MARIANA ISLANDS	MP
NORWAY	NO
OMAN	OM
PAKISTAN	PK
PALAU	PW
PALESTINIAN TERRITORY, OCCUPIED	PS
PANAMA	PA
PAPUA NEW GUINEA	PG
PARAGUAY	PY
PERU	PE
PHILIPPINES	PH
PITCAIRN	PN
POLAND	PL

PORTUGAL	PT
PUERTO RICO	PR
QATAR	QA
RÉUNION	RE
ROMANIA	RO
RUSSIAN FEDERATION	RU
RWANDA	RW
SAINT HELENA	SH
SAINT KITTS AND NEVIS	KN
SAINT LUCIA	LC
SAINT PIERRE AND MIQUELON	PM
SAINT VINCENT AND THE GRENADINES	VC
SAMOA	WS
SAN MARINO	SM
SAO TOME AND PRINCIPE	ST
SAUDI ARABIA	SA
SENEGAL	SN
SERBIA AND MONTENEGRO	CS
SEYCHELLES	SC
SIERRA LEONE	SL
SINGAPORE	SG
SLOVAKIA	SK
SLOVENIA	SI

SOLOMON ISLANDS	SB
SOMALIA	SO
SOUTH AFRICA	ZA
SOUTH GEORGIA AND THE SOUTH SANDWICH ISLANDS	GS
SPAIN	ES
SRI LANKA	LK
SUDAN	SD
SURINAME	SR
SVALBARD AND JAN MAYEN	SJ
SWAZILAND	SZ
SWEDEN	SE
SWITZERLAND	CH
SYRIAN ARAB REPUBLIC	SY
TAIWAN, PROVINCE OF CHINA	TW
TAJIKISTAN	TJ
TANZANIA, UNITED REPUBLIC OF	TZ
THAILAND	TH
TIMOR-LESTE	TL
TOGO	TG
TOKELAU	TK
TONGA	TO
TRINIDAD AND TOBAGO	TT

TUNISIA	TN
TURKEY	TR
TURKMENISTAN	TM
TURKS AND CAICOS ISLANDS	TC
TUVALU	TV
UGANDA	UG
UKRAINE	UA
UNITED ARAB EMIRATES	AE
UNITED KINGDOM	GB
UNITED STATES	US
UNITED STATES MINOR OUTLYING ISLANDS	UM
URUGUAY	UY
UZBEKISTAN	UZ
VANUATU	VU
Vatican City State - refer to HOLY SEE	VA
VENEZUELA	VE
VIET NAM	VN
VIRGIN ISLANDS, BRITISH	VG
VIRGIN ISLANDS, U.S.	VI
WALLIS AND FUTUNA	WF
WESTERN SAHARA	EH
YEMEN	YE
Zaire - refer to CONGO, THE DEMOCRATIC REPUBLIC	CG

OF THE

ZAMBIA

ZM

ZIMBABWE

ZW