

# World Pay App



## Purpose

This Acquire page app allows you to create an Electronic Point of Sale project, to take payment from a pin pad card reader, using the Worldpay (formally Yespay) system.

It connects to the Worldpay EasyVTerminal software which interfaces with an attached card reader.

You can show an image while the user follows the instructions on the card reader.

Actions can be performed on Events. E.g. Set Acquire Variables, jump to a page etc. on a successful or failed transaction or on an error.

Fully configurable, itemized receipts can be printed.

## Requirements

The Worldpay EasyVTerminal software must be installed, licensed and configured on each of your player machines.

Be sure to configure the software to:

- Run in the Kiosk Environment.
- Use the Socket interfacing method.
- Have a reasonable Card inactivity Timeout value.
- Position the window off-screen. This will ensure that users do not see the EasyVTerminal during a payment request.

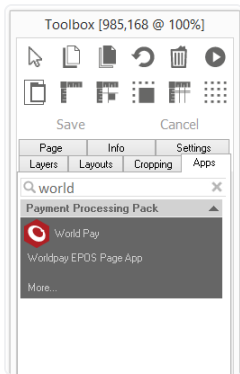
Please refer to the Worldpay documentation for more information.

as:lightblue This app was written for and tested with EasyVTerminal v2.2.4.2

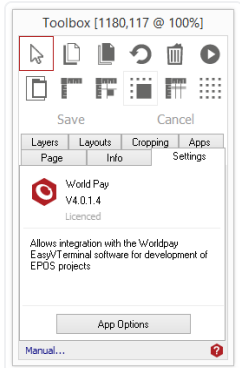
## Using the WorldPay App.

To access the app open an existing or new page and select the **'App'** tab of the page editor **'Toolbox'** (shown). Apps are grouped within the relevant app packs and are displayed red if they have been licensed. For more information about licensing apps go to [Licensing apps](#).

Select the app from the list or search for it using the search bar. When an app is selected it will be highlighted as shown.

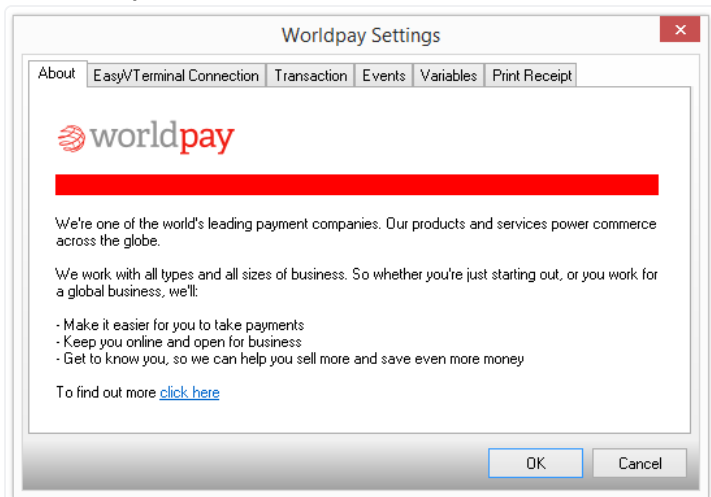


Once you have selected your app draw an area on your page where you would like to use the app. The **'Settings'** tab within the Toolbox will automatically open when the app is added to the page. From here you can set your app up by clicking on the **'App Options'** button.



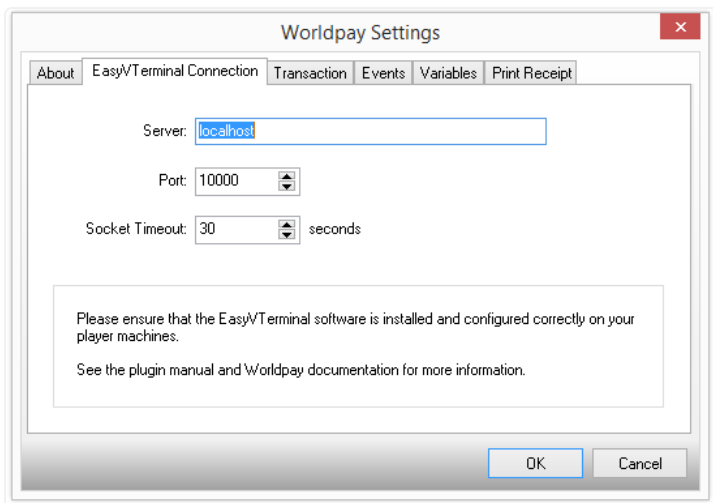
## About Tab.

This tab provides more information about the app and includes a link to the Worldpay website.



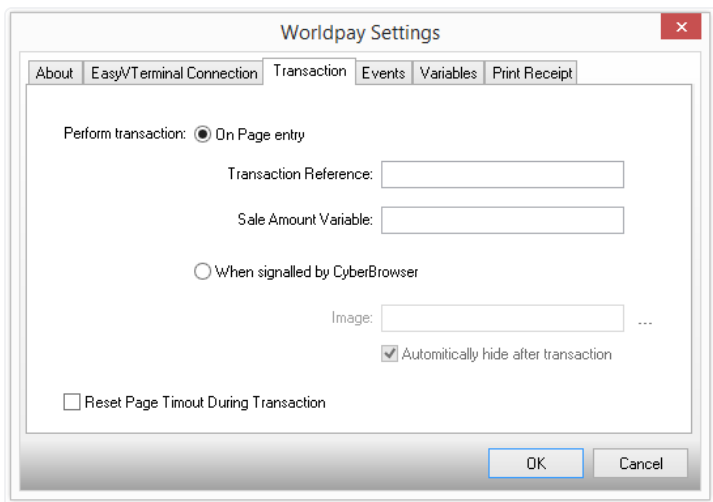
## EasyVTerminal Connection tab.

Use this tab to set up the connection with the EasyVTerminal software.



The **Server** will usually be localhost, if not please refer to the EasyVTerminal documentation.  
**Port:** must match the one in the EasyVTerminal configuration.

### Transaction tab.



Request payment when the Acquire Page is entered:

- A Transaction Reference to be used for tracking transactions.
- The amount to be requested should be stored in the Sale Amount Variable. This would typically have been set on a previous page, from an edit box for example.

See Appendix I for how to use this method.

**When signaled:** A website can signal the Worldpay app to request a payment.

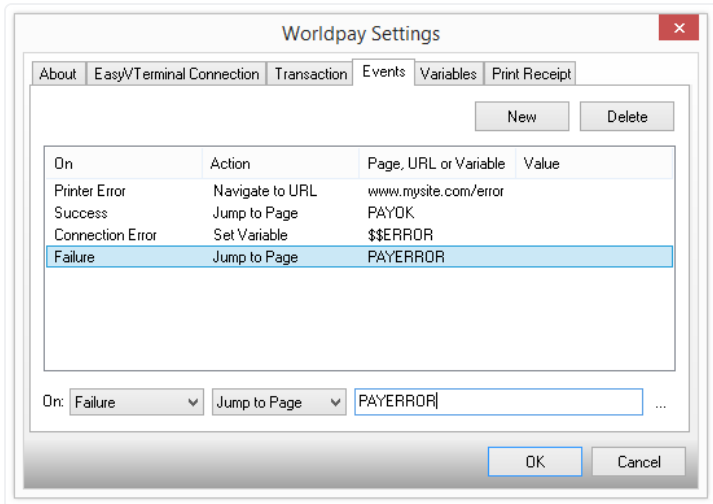
- The app will show the Image entered while processing the payment.
- You can set the image to automatically hide after the transaction. Otherwise the user must click the image to hide it.

as:lightbulb image can be an Acquire Variable. If so, as your project does not reference them directly you must place these images inside an “Always Send” folder in your media folder to ensure they get deployed with your project.

see Appendix II for more detail.

### Events tab.

Use this tab to set Actions to perform on Events. This way you can present information to the user during payment, on a successful or failed transaction and on errors.



**New/Delete:** Use these buttons to add and remove events.

A **list** of events and actions to perform.

**On:** The event and action of the item selected in the above list

**A: Select a transaction event:**

- i. Request – Action is performed just before the transaction is requested.
- ii. Success – Action is performed upon a successful transaction.
- iii. Failure – Action is performed upon a failed transaction.
- iv. Printer Error – Action is performed when the printer is found not to be working. The printer is checked when the Acquire Page is first loaded and whenever a receipt is about to be printed.
- v. Connection Error – Action is performed when there is no connection with the EasyVTerminal (e.g. it is not running). Connection is tested when the Acquire Page is first loaded and at a transaction.

**B: Select an Action to perform on the Event**

- i. Jump to Page – Enter the Page ID to jump to.
- ii. Set Variable – Enter a Variable name and value.
- iii. Navigate to URL – Enter a URL, if CyberBrowser is on same page it will navigate to this URL. See Appendix II.

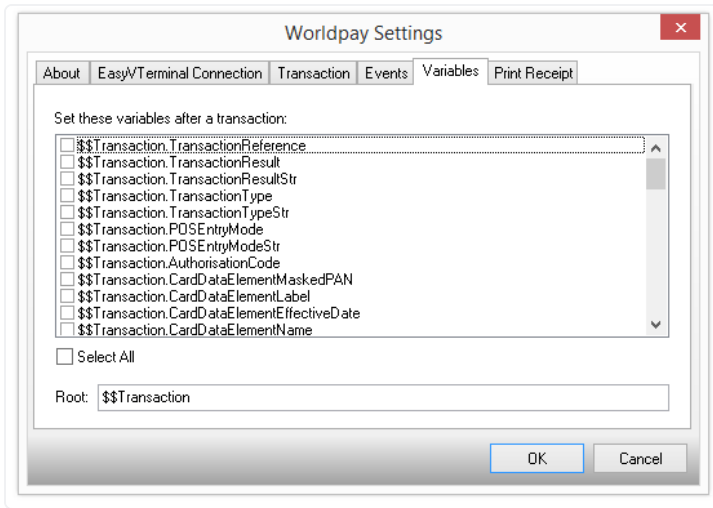
**C: Use the boxes to set the parameters for that Action (see b.)**

**as:lightbulb** URLs can contain Acquire Variables (see below) to pass the transaction reference/result etc. to your web page.

**as:lightbulb** The Events are processed from the top of the list to the bottom in two passes. First all variables action Events are processed to ensure that they are ready to be used in your jump actions. Then the jump action Events are processed.

**Variables tab.**

Use this tab to select what transaction result data you would like to save to Acquire Variables. These variables can then be used to present information to the user e.g. on an Acquire results page or by passing them back in the URL a website page.



## Print Receipt tab.

There are 3 methods of printing a receipt.

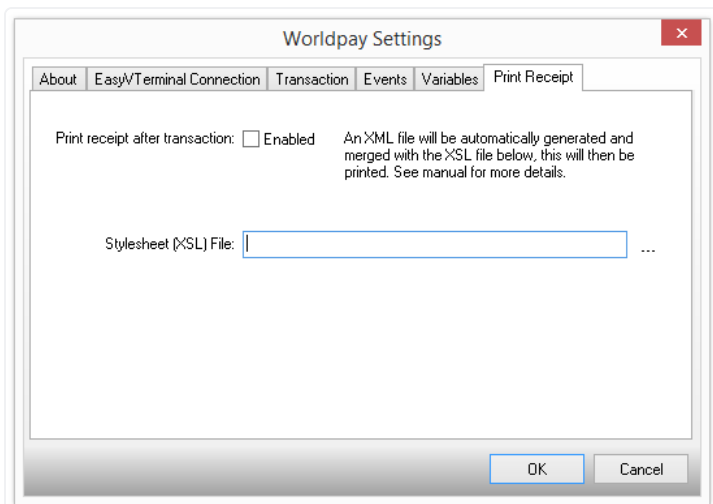
1. Let EasyVTerminal print the receipt. This will result in a non-configurable, non-itemized standard receipt. See your Worldpay documentation.
2. Set Variables in this app and jump to a page after a transaction which contains the Generic Printer app. Refer to that apps manual for how to use the variables to print a customizable receipt. This method does not support itemized receipts.
3. Let this Worldpay app print the receipt. This method is fully configurable and supports itemized receipts. See below.

Use this tab to select if and how you would like to print a receipt for a transaction. Receipt printing works by supplying the relevant receipt data in XML format. This can be generated automatically (See Appendix IV) or passed via from your website (See Appendces II and V).

The XML data is then formatted into a printable receipt using an XSL Style Sheet.

1. If this is enabled an XML file will be generated automatically following a transaction.
2. Enter the name of an existing Style Sheet. This is used to format your XML data into a printable receipt. See examples in the appendices.

Receipt printing will occur if a valid XSL file is found. If you do not wish the Worldpay app to print a receipt leave this blank.



## Appendix I – On Page Entry mode

Use this mode when taking payment directly from your project. You create a project to jump different pages to take a payment and display success or failure.

Example project description

- Give the user some way of selecting an amount and set that amount to an Acquire Variable.
- When ready jump to a **“payment”** page containing the Worldpay app.
- The payment page can contain a message such as **“please follow the pinpad instructions”**.
- Configure the Worldpay app to perform the transaction On Page Entry and to use the previously set amount variable.
- Configure the Worldpay app to jump to pages on transaction Success and Failure.
- You can also set variables to display messages on those windows e.g. if the pages are the same **“result”** page and you just want to show a success/fail message.
- Configure the Worldpay app to jump to pages printer and connection errors.
- Enter a receipt XSL and Enable automatic to print out a simple receipt.
- See Appendix III for a list of transaction results.

## Appendix II

Use this mode when you require a website to request payment. The Worldpay app must be on the same page. Your website, loaded into **CyberBrowser**, can send special Java Script commands (see below). **CyberBrowser** then tells the Worldpay app to display itself and take payment or print a receipt etc.

### Example project description:

- Place a **CyberBrowser** app on the page and configure it to load your website and accept Java Script commands from it.
- Place a Worldpay app over the top of **CyberBrowser**. It is transparent and won't interfere with the page in any way.
- Configure the Worldpay app to perform a transaction when signaled by **CyberBrowser**.
- Your website calls the special Java Script command **cb\_makeWorldpaypayment** (see section below) passing a transaction reference and amount.

The Image will be shown during the transaction. Your page must now wait until the transaction completes and then alert the user.

You can do this is various ways:

- Make the Image an Acquire Variable and set that to a fail/success image in the events tab.
- Get **CyberBrowser** to tell your webpage the result via the special callback function **WorldpayTransactionResult** on your page (see section below) and show results there.
- Get **Cyberbrowser** to navigate to a fail/success URL using the Events tab.

The transaction reference and result can be added to the URL's paramteters using the Acquire Variables

e.g.:

www.paid.html?transref=\$\$transaction.transactionreference&transres=\$\$transactionresult

as:lightbulb Note: Acquire Variables in the URL must be enabled in the Variables tab.

- Your website can call `cb_printWorldpayreceipt` after a transaction. See below.
- Your website can call `cb_getWorldpaystatus` prior to a transaction. See below.

See Appendix III for a list of transaction results.

### The special Java Script commands

Cyberbrowser provides a special Java Script command to request a payment from within your web page. This will only work if the CyberBrowser and Worldpay apps are on the same Acquire Page. CyberBrowser must also be configured to accept commands from your web site. There are 3 Commands:

`window.external.cb_getWorldpaystatus()`

Use this function to determine the working status of the Worldpay app before proceeding with a transaction etc.

Result A status bitmask where bit:

- 1 = Unable to get status
- 2 = Unable to connect to the EasyVTerminal
- 4 = There is a printer error

`window.external.cb_makeWorldpaypayment(reference, amount, callback)`

Tell CyberBrowser to request a payment via the Worldpay app.

Reference	A transaction reference string. This will also be passed in the callback function <code>worldpayTransactionResult</code> (see below) parameters.
Amount	The payment amount to request as a string e.g. "10.99".
Callback	Boolean. If TRUE then <code>worldpayTransactionResult()</code> will be called on your webpage once the transaction is complete. See below.
Result	Boolean. TRUE if the request was made or FALSE if not. <b>Note:</b> This is <i>not</i> the result of the transaction. Your page should now wait until the transaction is complete (failed or successful). To obtain the result you should use one of the methods outlined above.

`window.external.cb_printWorldpayreceipt(reference, xml)`

Tell CyberBrowser to print a receipt via the Worldpay app.

reference	Your transaction reference for the receipt.
xml	All the data you wish to print in xml format. See Appendix V for an example.
result	TRUE if receipt was printed or FALSE if not.

If callback was set to TRUE in `cb_makeWorldpaypayment` then the following function will be call on your webpage:

WorldpayTransactionResult(reference, result)

Reference: The transaction reference that was passed to cb\_makeWorldpaypayment.

Result: The result of the transaction. See Appendix III for a list.

as:lightbulb **Note!** Ensure that this function is on the same page that calls cb\_makeWorldpaypayment and do not navigate away until it is called.

Examples:

1. if (window.external)

```
    if (window.external.cb_version) {  
        status = window.external.cb_getWorldpaystatus();  
        do something depending on status errors  
    }
```

2. if (window.external)

```
    if (window.external.cb_version)  
        if window.external.cb_makeWorldpaypayment('1234', '9.99',  
true)  
  
        wait for transaction result...  
    else  
        show failed error...  
  
    - CyberBrowser/Worldpay apps not on page or  
    - Worldpay software not running/licenced/connected or  
    - Worldpay app not set to CyberBrowser mode
```

3. if (window.external)

```
    if (window.external.cb_version)  
        if window.external.cb_printWorldpayreceipt('1234', xml)  
            do something on success  
        else  
            do something on fail
```

Where `xml` = is a string containing your actual data in xml format. See Appendix V.

4. Your callback function should be defined like this:

```
function WorldpayTransactionResult(Reference, Result) {  
    do something with the result e.g. print a receipt!  
}
```

This function is called after a transaction as a result of the above cb\_makeWorldpaypayment.

### **Appendix III – Transaction Result Codes**

These code numbers are used to set the TransactionResult variable and are passed to WorldpayTransactionResult.

#### **Success...**

- 1 - Approved Online
- 2 - Approved Offline
- 3 - Approved Manual (Referral)

#### **Failure...**

- 4 - Declined Online
- 5 - Declined Offline
- 9 - Cancelled 1
- 6 - Capture Card, declined online
- 19 - Transaction Aborted
- 20 - Pre sales completed
- 21 - Pre sales rejected
- 22 - Card number not matched
- 23 - Expiry date not matched
- 24 - Invalid transaction state
- 25 - Transaction not valid for requested operation
- 26 - Invalid PGTR
- 27 - Invalid Merchant
- 28 - Invalid Terminal
- 29 - Merchant status is not valid
- 30 - Invalid card number
- 31 - Expired Card
- 32 - Pre valid card
- 33 - Invalid issue number
- 34 - Invalid card expiry date
- 35 - Invalid start date
- 36 - Card not accepted
- 37 - Transaction not allowed
- 38 - Cash back not allowed
- 42 - Status Busy
- 43 - Status Not Busy
- 50 - AVS details required
- 60 - Card activated (Gift card activation)
- 61 - Card already activated (Gift card activation)
- 62 - Card not activated (For Gift Card)
- 63 - Hot Listed Card (for Gift Card)
- 64 - Insufficient Balance (for Gift Card)
- 65 - Invalid Security Code (for Gift Card)

#### **Connection to EasyVTerminal Failure Etc...**

- 2000 - Failed To Send
- 2001 - Socket Timeout Reading Response
- 2002 - Socket Error Reading Response
- 2003 - Request Failed As Busy

2004 - Request Failed To Connect

2005 - Invalid Amount

### Other...

-1 - A null result. Probably due to transaction timeout

## Appendix IV – Automatic Receipt XML and example XSL

**Important Note:** The XML, XLS and Receipt below are for example only. They do not necessarily conform to any legal requirements of printing a receipt.

When “Print receipt after transaction” is enabled the following XML is automatically generated after a transaction.

```
<receipt>
<transactiondate>20/07/2012</transactiondate>
<transactiontime>16:13:56</transactiontime>
<receiptnumber>XXX</receiptnumber>
<transactionreference>ref4321</transactionreference>
<totalsaleamount>0.01</totalsaleamount>
<totalamount>0.01</totalamount>
<transactionerror>Cancelled [9]</transactionerror>
<transactionerrorstr>TRANSACTION VOID</transactionerrorstr>
</receipt> To format this into a receipt your XSL Style Sheet might look something like this: <xsl:stylesheet
version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform"> <xsl:template match="/">
<html>
<style>
body, table, tr, td { background: #FFFFFF; color: #000000; font-family: Arial, Verdana, Helvetica, sans-serif; font-
size: 10pt; font-weight: normal; line-height: 16px; };
</style>
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0" text="black" color="white">
CUSTOMER RECEIPT<br /> ACQUIRE<br /> 190 Scudamore Road<br /> Leicester<br /> LE3 1UQ<br /> UK<br />
<xsl:copy-of select="receipt/transactiondate"/> <xsl:copy-of select="receipt/transactiontime"/><br />
RECIEPT NO.:<xsl:copy-of select="receipt/receiptnumber"/><br /> TXN REF.:<xsl:copy-of
select="receipt/transactionreference"/><br />
<table border="0" cellpadding="2" cellspacing="1" width="100%"> <tr> <td align="left">SALE:</td> <td
align="right"><xsl:copy-of select="receipt/totalsaleamount"/> </td>
</tr> <tr> <td align="left">TOTAL: </td> <td align="right"><xsl:copy-of select="receipt/totalamount"/> </td>
</tr>
</table>
<xsl:if test="receipt/transactionerror != ""> <xsl:copy-of select="receipt/transactionerror"/><br /> <xsl:copy-of
select="receipt/transactionerrorstr"/><br /> </xsl:if>
</body>
</html>
</xsl:template>
</xsl:stylesheet>
```

**Which would result in a receipt like this:**

```

CUSTOMER RECEIPT
ACQUIRE
190 Scudamore Road
Leicester
LE3 1UQ
UK
20/07/2012 16:13:56
RECEIPT NO.:XXX
TXN REF.:ref4321
SALE:                0.01
TOTAL:               0.01
Cancelled [9]
** TRANSACTION VOID **

```

## Appendix V – Example cb\_printWorldpayreceipt XML/XSL

as:lightblue The XML, XLS and Receipt below are for example only. They do not necessarily conform to any legal requirements of printing a receipt.

When you call the cb\_printWorldpayreceipt Java Script command you pass it your own receipt data as XML. This way you can create an itemized receipt. E.g.:

```

if (window.external)
if (window.external.cb_version)
if (!window.external.CB_PRINTWORLDPAYRECEIPT('1234',
'<receipt>'+
'<data>'+
'<address>'+
'<line>Fruit Basket</line>'+
'<line>1 Lime Street</line>'+
'<line>Lemonton Spa LE1 0AB</line>'+
'<line>01234 567890</line>'+
'<line>Shop 24/7 online at</line>'+
'<line>www.fruitbasket.web</line>'+
'</address>'+
'<date>05/07/12 17:16</date>'+
'<slip>3010042433</slip>'+
'<staff>Matt</staff>'+
'<trans>1234</trans>'+
'<barcode>9876543210</barcode>'+
'<gbp>5.48</gbp>'+
'<cash>5.48</cash> '+
'<returns>'+
'<heading>Returns policy</heading>'+
'<policy>'+
'If you are not entirely satisfied with '+
'your purchase, please return the goods '+
'within 14 days of purchase. Provided '+
'the goods are in their original packaging '+
'and you have proof of purchase, then '+

```

```
'we will exchange the goods or issue '+
'a credit note for use within 12 months.'+
'</policy>'+
'</returns>'+
'<vat>'+
'<field1>C</field1>'+
'<field2>20%</field2>'+
'<field3>0.00</field3>'+
'<field4>0.00</field4>'+
'</vat>'+
'</data>'+
'<items>'+
'<item>'+
'<field1>Oranges</field1>'+
'<field2>1.99</field2>'+
'<field3>a</field3>'+
'</item>'+
'<item>'+
'<field1>Strawberries</field1>'+
'<field2>3.49</field2>'+
'<field3></field3>'+
'</item>'+
'</items>'+
'</receipt>') )
alert('Print request failed');
```

**To format this into a receipt your XSL Style Sheet might look something like this:**

```
xsl:stylesheet version="1.0" xmlns:xsl="http://www.w3.org/1999/XSL/Transform">
<xsl:template match="/">
<html>
<style>
body, table, tr, td {
background: #FFFFFF;
color: #000000;
font-family: Arial, Verdana, Helvetica, sans-serif;
font-size: 10pt;
font-weight: normal;
line-height: 16px;
};
</style>
<body leftmargin="0" topmargin="0" marginwidth="0" marginheight="0" text="black"
color="white">
<br />
<!-- image -->
<div align="center">

```

```

</div> <br />
<!-- address -->
<div align="center">
<xsl:for-each select="receipt/data/address/line">
<xsl:copy-of select="."/;><br/>
</xsl:for-each>
<hr style="border:dashed black; border-width:1px 0 0; height:0;"></hr>
</div>
<!-- info -->
<table border="0" cellpadding="2" cellspacing="1" width="100%">
<tr>
<td align="left">Date: </td>
<td align="left"><xsl:copy-of select="receipt/data/date"/></td>
<td align="left">Slip: </td>
<td align="right"><xsl:copy-of select="receipt/data/slip"/></td>
</tr>
<tr>
<td align="left">Staff: </td>
<td align="left"><xsl:copy-of select="receipt/data/staff"/></td>
<td align="left">Trans: </td>
<td align="right"><xsl:copy-of select="receipt/data/trans"/></td>
</tr>
</table>
<hr style="border:dashed black; border-width:1px 0 0; height:0;"></hr>
<!-- items titles -->
<table border="0" cellpadding="2" cellspacing="1" width="100%">
<tr>
<td align="left">Description</td>
<td align="right">Amount</td>
</tr>
</table>
<hr style="border:dashed black; border-width:1px 0 0; height:0;"></hr>
<!-- barcode -->
<table border="0" cellpadding="2" cellspacing="1" width="100%">
<tr> <td align="left">Barcode:</td>
<td align="left"><xsl:copy-of select="receipt/data/barcode"/></td>
</tr>
</table>
<!-- items -->
<table border="0" cellpadding="2" cellspacing="1" width="100%">
<xsl:for-each select="receipt/items/item">
<tr>
<td align="left"><xsl:copy-of select="field1"/></td>

```

```

<td align="right"><xsl:copy-of select="field2"/></td>
<td align="right" width="10"><xsl:copy-of select="field3"/></td>
</tr>
</xsl:for-each>
</table>
<hr style="border:dashed black; border-width:1px 0 0; height:0;"></hr>
<!-- totals --><table border="0" cellpadding="2" cellspacing="1" width="100%" >
<tr>
<td align="left">Total GBP</td><td align="right">
<xsl:copy-of select="receipt/data/gbp"/>
</td>
</tr>
<tr>
<td align="left">Cash</td> <td align="right"><xsl:copy-of select="receipt/data/cash"/></td>
<td align="right" width="10"></td>
</tr>
</table>
<hr style="border:dashed black; border-width:1px 0 0; height:0;"></hr>
<!-- VAT -->
<table border="0" cellpadding="2" cellspacing="1" width="100%">
<tr>
<td align="left">VAT</td>
<td align="right"><xsl:copy-of select="receipt/data/vat/field1"/></td>
<td align="right"><xsl:copy-of select="receipt/data/vat/field2"/></td>
<td align="right"><xsl:copy-of select="receipt/data/vat/field3"/></td>
<td align="right"><xsl:copy-of select="receipt/data/vat/field4"/></td>
</tr>
</table> <hr style="border:dashed black; border-width:1px 0 0; height:0;"></hr>
<!-- returns policy -->
<div align="center"> <xsl:copy-of select="receipt/data/returns/heading"/><br/>
<xsl:copy-of select="receipt/data/returns/policy"/>
</div>
</body>
</html>
</xsl:template>
</xsl:stylesheet>

```

**Which would result in a receipt like this:**



Fruit Basket  
1 Lime Street  
Lemonton Spa LE1 0AB  
01234 567890  
Shop 24/7 online at  
[www.fruitbasket.web](http://www.fruitbasket.web)

Date: 05/07/12 17:16 Slip: 3010042433  
Staff: Matt Trans: 1234

Description	Amount
Barcode: 9876543210	
Oranges	1.99 a
Strawberries	3.49
Total GBP	5.48
Cash	5.48

VAT C 20% 0.00 0.00

#### Returns policy

If you are not entirely satisfied with your purchase, please return the goods within 14 days of purchase. Provided the goods are in their original packaging and you have proof of purchase, then we will exchange the goods or issue a credit note for use within 12 months.

## Appendix VI – Logging Codes

Transactions etc. are logged to the Acquire log files. These can be used to trace transactions and be alerted via the Plus! Network of faults etc.

### Here is a list of the message codes logged:

#### Transactions

- LOG\_PAYMENTSUCCESS: 711 Contains Transaction details. †
- LOG\_PAYMENTFAILURE: 712 Contains Transaction details. †
- LOG\_PAYMENTPROCESSING: 713 Various information. Contains Transaction details. †

#### Receipt Printing

- LOG\_PRINTINGOK 709
- LOG\_PRINTINGFAILED 710

#### Connection Errors

- LOG\_CONNECTIONERROR 707

#### Printer Errors

- LOG\_PRINTERPAPERJAM 137
- LOG\_PRINTERPAPERISSUE 138
- LOG\_PRINTEROUTPUTBINFULL 139
- LOG\_PRINTERNOTFOUND 140
- LOG\_PRINTERNOTONER 142

LOG\_PRINTERPAGEPUNT 143  
LOG\_PRINTERUSERINTRVNTN 144  
LOG\_PRINTEROUTOFMEMORY 145  
LOG\_PRINTERDOOROPEN 146  
LOG\_PRINTERPAPEROUT 150  
LOG\_PRINTEROFFLINE 152  
LOG\_PRINTERERROR 153

† Logged Transaction details may contain the transaction reference and amount etc. but is otherwise anonymous. No customer data what-so-ever is logged.