



PC2Paper Letter API Interfaces

V3.2

Contents

Contents	2
Introduction	3
The PC2Paper Form Post Interface	4
Why or when should I use it?	4
How it works	4
HTTP Post Fields.....	4
Command Fields.....	4
Message Body Fields	5
Sender Fields.....	5
Postage Fields	5
Example HTML for an HTTP Post	6
Example ASP code for backend server posting.....	8
Example Delphi Code	9
PC2Paper XML RPC Interface.....	10
Why or when should I use it?	10
Where Can I find more help?	12
Error Codes	12
Appendix A	13
Country Codes	13

Introduction

The PC2Paper Letter API Interfaces enables the back end sending of letters to PC2Paper.

The benefits of this approach are unlimited. If you integrate this interface with your website PC2Paper can automatically send letters or invoices to your clients once they have purchased items from your website. Integrating PC2Paper with any back end system can provide a seamless fully automated system removing the cost of an internal mailing department in your organisation to send out letters.

PC2Paper's API interfaces provide a feedback mechanism to your web server to inform it of the status of your letters.

We currently offer two API interfaces for sending letters both of these interfaces use an HTTP post of information between our servers and yours.

These interfaces are:

- XML RPC interface (Only used for PDF and Word documents)
- Form Post Interface (FPI) (Sending of plain text and HTML only)

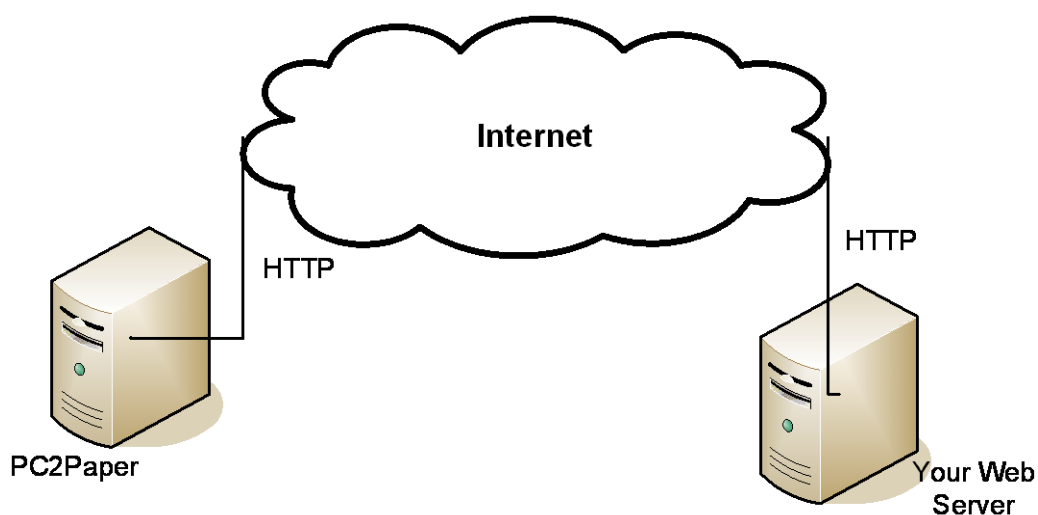
The PC2Paper Form Post Interface

The PC2Paper Form Post Interface is the simplest of our interfaces and most favoured by our users. It enables the sending of letters in plain text and HTML only and meets most of our customer's requirements.

Why or when should I use it?

The Form Post Interface should be used if you would like to send only HTML or text letters. For example you may want to send letters that are automatically constructed on your server or in your application that contain a letter head or signature. You could construct these letters in simple HTML and embed images in your letter that are available from a public server.

How it works



The Form Post Interface workings are quite simple. Your server provides an HTTP post to our servers with the following HTTP fields.

HTTP Post Fields

When sending a letter the system expects the following fields to be sent.

Command Fields

Field	Description
cmd	This should be set to SendLetter when sending letters via the interface.
login	Your login name
password	Your password

blnDebug	Set this to true if you are having problems with the interface and wish to see error descriptions.
blnIncludeSenderAddress	Set to true if you wish to include the senders address on the letter

Message Body Fields

Field	Description
strName	The name of the receiver
strHouseName	The house number or name of the receiver
strStreet	The street name of the receivers address
strTownCity	The receivers town or city
strCounty	The receivers county or state
strPostCode	The receivers postcode
iCountry	The receivers country number represented as an integer (refer to Appendix A)
strBody	The body text of the letter

Sender Fields

Field	Description
strSenderName	The name of the receiver
strSenderHouseName	The house number or name of the receiver
strSenderStreet	The street name of the receivers address
strSenderTownCity	The senders town or city
strSenderCounty	The senders county or state
strSenderPostCode	The senders postcode
iSenderCountry	The senders country number represented as an integer (refer to Appendix A)

Postage Fields

The postage fields are used by the API to determine where your letter will be sent from, if you requested extra features such as recorded delivery with your letter, the paper your letter will be printed on and the envelope your letter will be sent in (etc DL). The site uses these options to work out the total cost of your letter, if you would like to know the total cost of your letter before sending it. Please refer to the documentation provided by our Letter Pricing API.

Field	Description
postage	This is the id of the printing station the letter is sent from (see Appendix B)
Extras	This is a list of features available from the printing station. This could be things such as Recorded Delivery. (see Appendix C)
Paper	This is the id of the type of paper the letter will be printed on. This can only be a paper

	type supported by the selected printing station. (see Appendix D)
Envelope	This is the id of the envelope type supported by the printing station. (see Appendix E)

Example HTML for an HTTP Post

The following HTML example can be used to test the interface from a traditional web browser.

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0 Transitional//EN">
<html>
  <head>
    <title>PC2Paper HTTP Post Example</title>
  </head>
  <body>
<form name="test" action="http://www.pc2paper.co.uk/remote/clienttalkmulti.asp">
  <b>Command Details</b><br>
  Login <INPUT type="text" NAME="login"><br>
  Password<INPUT type="text" NAME="password"><br>
  Command<INPUT type="text" NAME="cmd" value="Sendletter"><br>
  Debug<input type="text" name="blnDebug" value="true"><br>
  <br>
  <b>Receivers Address Details</b><br>
  strName<INPUT type="text" NAME="strName"><br>
  strHouseName<INPUT type="text" NAME="strHouseName"><br>
  strStreet<INPUT type="text" NAME="strStreet"><br>
  strTownCity<INPUT type="text" NAME="strTownCity"><br>
  strCounty<INPUT type="text" NAME="strCounty"><br>
  strPostCode<INPUT type="text" NAME="strPostCode"><br>
  iCountry<INPUT type="text" NAME="iCountry"><br>
  <br>
  <b>Senders Address Details</b><br>
Set the blnIncludeSenderAddress to true if you wish to include the senders
address.
  <br>blnIncludeSenderAddress<input type="text" name="blnIncludeSenderAddress"
value="true"><br>
  <b>Postage</b><br>
  strPostage <INPUT type="text" NAME="postage" value="3"><br>
  strExtras <INPUT type="text" NAME="Extras" value=""><br>
  strPaper <INPUT type="text" NAME="Paper" value="1"><br>
  strEnvelope <INPUT type="text" NAME="Envelope" value="1">
  <br><br>
  strSenderName<INPUT type="text" NAME="strSenderName"><br>
  strSenderHouseName<INPUT type="text" NAME="strSenderHouseName"><br>
  strSenderStreet<INPUT type="text" NAME="strSenderStreet"><br>
  strSenderTownCity<INPUT type="text" NAME="strSenderTownCity"><br>
  strSenderCounty<INPUT type="text" NAME="strSenderCounty"><br>
  strSenderPostCode<INPUT type="text" NAME="strSenderPostCode"><br>
  iSenderCountry<INPUT type="text" NAME="iSenderCountry"><br>
  strBody<INPUT type="text" NAME="strBody"><br>
  <br>
  <INPUT type="submit" value="Submit" ID="Submit1" NAME="Submit1">
</form>
  </body>
</html>

```

Please note: The code above should only be used for testing purposes, and never from your front end website as it exposes your login name and password in the HTML code.

Example ASP code for backend server posting

The following example uses MSXML 2.6 and creates a backend posting from an ASP server to the PC2Paper Server.

```
<%
'-----
' Example provided by PC2Paper
' You will require a copy of MSXML in order to use this example
' Example is intended for Microsoft ASP based websites
'-----

str=AppendString("strName",strName) & "&" &
AppendString("strHouseName",strHouseName) & "&"

str=str & AppendString("strHouseName",strHouseName) & "&" &
AppendString("strStreet",strStreet) & "&"
str=str & AppendString("strTownCity",strTownCity) & "&" &
AppendString("strCounty",strCounty) & "&"
str=str & AppendString("strPostCode",strPostCode) & "&" &
AppendString("iCountry",iCountry) & "&"
str=str & AppendString("strBody",strBody) & "&"

str=str & AppendString("strSenderName",strSenderName) & "&" &
AppendString("strSenderHouseName",strSenderHouseName) & "&"
str=str & AppendString("strSenderStreet",strSenderStreet) & "&" &
AppendString("strSenderTownCity",strSenderTownCity) & "&"
str=str & AppendString("strSenderCounty",strSenderCounty) & "&" &
AppendString("strSenderPostCode",strSenderPostCode) & "&"
str=str & AppendString("strSenderPostCode",strSenderPostCode) & "&" &
AppendString("iSenderCountry",iSenderCountry) & "&"

str=str & AppendString("postage","1") & "&"
str=str & AppendString("Extras","0") & "&"
str=str & AppendString("paper","1") & "&"
str=str & AppendString("envelope","1") & "&"

str=str & AppendString("cmd","SendLetter") & "&"
str=str & AppendString("login","MyUsername") & "&"
str=str & AppendString("password","MyPassword") & "&"
str=str & AppendString("blnIncludeSenderAddress","true") & "&"
str=str & AppendString("blnDebug","false")

response.Write(SendLetter(str))

function AppendString(strField,sValue)
    AppendString = strField & "=" & sValue
end function

function SendLetter(sendstring)
    set objHttp = Server.CreateObject("Msxml2.ServerXMLHTTP")
    objHttp.open "POST",
"https://www.pc2paper.co.uk/remote/clienttalkmulti.asp", false
    objHttp.Send sendstring

    SendLetter = objHttp.responseText
end function
%>
```


Example Delphi Code

The Delphi version below uses the Indy 10 library. Please note that we have not used the `TIdMultiPartFormDataStream` method as it stopped working from Indy version 9 onwards and was still not working when we tested it with Indy 10. Please let us know if you get better results than we did with `TIdMultiPartFormDataStream`.

```
var
  PostData: TStringList;
  PostDataStream : TStringStream;
  Results : string;
  IdHTTP1: TIdHTTP;

begin
  IdHTTP1.HandleRedirects :=false;
  IdHTTP1.Request.ContentType := 'application/x-www-form-urlencoded';

  PostDataStream := TStringStream.Create("");

  PostData := TStringList.Create;

  PostData.Values['cmd'] := SendLetter;
  PostData.Values['login'] := 'pc2paperusername';
  PostData.Values['password'] := 'pc2paperpassword';
  PostData.Values['strName'] := 'Bob Smith';
  PostData.Values['strStreet'] := '10 My Street';
  PostData.Values['strTownCity'] := 'My Town';
  PostData.Values['strCounty'] := 'County or State';
  PostData.Values['strPostCode'] := 'Postcode or ZIP';
  PostData.Values['iCountry'] := '1'; //entry code found in appendix
  PostData.Values['strBody'] := 'Hi how are you doing?';
  PostData.Values['strSenderName'] := 'You';
  PostData.Values['strSenderHouseName'] := '10 Your Street';
  PostData.Values['strSenderTownCity'] := 'Your Town';
  PostData.Values['strSenderCounty'] := 'Your County or State';
  PostData.Values['strSenderPostCode'] := 'Your Postcode or ZIP';
  PostData.Values['iSenderCountry'] := '1'; //entry code found in appendix

  PostData.Values['postage'] := '1'; //See the Postage Fields section in the API document.
  PostData.Values['Extras'] := '0';
  PostData.Values['Paper'] := '1';
  PostData.Values['Envelope'] := '1';

  PostData.Values['bInDebug'] := 'false';
  PostData.Values['bInIncludeSenderAddress'] := 'true';
```

```

IdHTTP1.Post('http://www.pc2paper.co.uk/remote/clienttalkmulti.asp',PostData,PostDataStream);

Results := PostDataStream.DataString;

end;

```

PC2Paper XML RPC Interface

The PC2Paper XML RPC Interface works in almost exactly the same way as the PC2Paper FPI interface however all of the information is sent as an XML string.

Why or when should I use it?

You should use this interface if you intend to send a PDF document as a letter instead of plain HTML or Text.

How it works

The RPC interface quite simply sends your letter to us in an XML format. Details such as the recipient of the letter and the postage are stored in XML fields while the actual PDF of the letter is encoded in BASE64 code.

The XML below is a simple example of what this request looks like.

```

<xmlFileSend>
  <Client>My Software client v1.0 by Company plc</Client>
  <LOGINID>your pc2paper account username</LOGINID>
  <Password>your pc2paper password</Password>

  <ReceiverName>Bob Smith</ReceiverName>
  <ReceiverAddressLine1>ReceiverAddressLine1 </ReceiverAddressLine1>
  <ReceiverAddressLine2>ReceiverAddressLine2 </ReceiverAddressLine2>
  <ReceiverAddressTownCityOrLine3>ReceiverAddressTownCityOrLine3</ReceiverAddressTownCityOrLine3>
  <ReceiverAddressCountyStateOrLine4>ReceiverAddressCountyStateOrLine4</ReceiverAddressCountyStateOrLine4>
  <ReceiverCountryCode>ReceiverCountryCode This must be an integer</ReceiverCountryCode>
  <ReceiverAddressPostCode>ReceiverAddressPostCode</ReceiverAddressPostCode>

  <YourRef> Your own reference</YourRef>
  <Postage>1</Postage>
  <Paper>1</Paper>
  <Extras>0</ Extras>
  <Envelope>1</ Envelope >
  <Pages>1</Pages>
  <SenderAddress>22 Some Road
  some street
  </SenderAddress>
  <Base64Data FileName="11184467.pdf" dt:dt="bin.base64" xmlns:dt="urn:schemas-microsoft-com:datatypes">
    Base64 binary encoding of your PDF or word document
  </Base64Data />
</xmlFileSend>

```

This is printing station you wish to send your letter from (See Appendix B)

This is the id of the paper your letter will be printed on (See Appendix C)

The XML is sent to us as an HTTP Form Post at the following address

<https://www.pc2paper.co.uk/remote/filepostMulti.asp>

A successful submission would return the following XML response.

<pre><LetterStatus> <LetterID>1013</LetterID>" <LetterTo>Bob Smith</LetterTo>" <LetterCost currency="GBP">39</LetterCost> <FundsLeft currency="GBP">5.52</FundsLeft> <Status>OK</Status> <Err errors=0></Err> </LetterStatus></pre>	<p>← This is a unique letter id we use to identify your letter</p>
	<p>← This is the cost of your letter and the funds left in your PC2Paper account</p>

Where Can I find more help?

If you need more help please do not hesitate to contact PC2Paper further using the email address customerservice@pc2paper.co.uk also visit our website www.pc2paper.co.uk

Error Codes

The following are error codes returned by the API interface. Please note you will need to set the field `blnDebug` to true to return debug descriptions.

ER5: Login Failed

Returned when your login name and password were incorrect.

ER6: Submitted a field that should not have been empty

You will receive this error when you have submitted a field that is blank and should have contained a value. The error code should return the name of the field.

ER7: Field must be integer

You will receive this error when the field you submitted should have contained an integer value and did not. The error code should return the name of the field.

ER8: Field is not a recognised country id

You will receive this error code when the field should have contained a country id (please refer to Appendix A)

Appendix A

Country Codes

Below are the country codes expected by the country field in the PC2Paper API interface. You need to enter these so we can identify the countries correctly in our database. You can download a CSV these codes in CSV format at the following address <http://www.pc2paper.co.uk/country.csv>

Country	
Country ID	Country Name
1	United Kingdom
2	Abu Dhabi (use UAE)
3	Afghanistan
4	Ajman (use UAE)
5	Albania
6	Algeria
7	Andorra
8	Angola
9	Anguilla
10	Antigua & Barbuda
11	Argentina
12	Armenia
13	Aruba
14	Ascension
15	Australia
16	Austria (EU)
17	Azerbaijan
18	Azores (EU)
19	Bahamas
20	Bahrain
21	Balearic Islands (EU)
22	Bangladesh
23	Barbados
24	Belarus
25	Belau (Palau)
26	Belgium (EU)
27	Belize
28	Benin
29	Bermuda
30	Bhutan
31	Bolivia
32	Bosnia-Herzegovina
33	Botswana
34	Brazil

Country	
Country ID	Country Name
35	British Indian Ocean
36	British Virgin Islands
37	Brunei Darussalam
38	Bulgaria
39	Burkina Faso
40	Burundi
41	Cambodia
42	Cameroon
43	Canada
44	Canary Islands
45	Cape Verde
46	Cayman Islands
47	Central African Rep.
48	Chad
49	Chile
50	China (People's Republic)
51	Christmas Island
52	Cocos Island (Keeling Island)
53	Colombia
54	Comoros
55	Congo
56	Congo Dem. Rep.
57	Corsica (EU)
58	Costa Rica
59	Cote d'Ivoire
60	Croatia
61	Cuba
62	Curacao
63	Cyprus
64	Czech Republic
65	Denmark (EU)
66	Djibouti
67	Dominica
68	Dominican Republic

Country	
Country ID	Country Name
69	Dubai (use UAE)
70	East Timor
71	Ecuador
72	Egypt
73	El Salvador
74	Equatorial Guinea
75	Eritrea
76	Estonia
77	Ethiopia
78	Falkland Islands
79	Faroe Islands
80	Fiji
81	Finland (EU)
82	France (EU)
83	French Guiana
84	French Polynesia
85	French Southern & Antarctic Territories
86	French West Indies
87	Fujairah (use UAE)
88	Gabon
89	Gambia
90	Gaza & KhanYunis
91	Georgia
92	Germany (EU)
93	Ghana
94	Gibraltar
95	Greece (EU)
96	Greenland
97	Grenada
98	Guam
99	Guatemala
100	Guinea
101	Guinea-Bissau
102	Guyana
103	Haiti
104	Honduras
105	Hong Kong
106	Hungary
107	Iceland
108	India
109	Indonesia
110	Iran
111	Iraq
112	Irish Republic (EU)

Country	
Country ID	Country Name
113	Israel
114	Italy (EU)
115	Jamaica
116	Japan
117	Jordan
118	Kazakhstan
119	Kenya
120	Kirghizstan
121	Kiribati
122	Korea (Democratic People's Republic)
123	Korea (Republic of)
124	Kuwait
125	Laos
126	Latvia
127	Lebanon
128	Lesotho
129	Liberia
130	Libya
131	Liechtenstein
132	Lithuania
133	Luxembourg (EU)
134	Macao
135	Macedonia (Former Yugoslav Rep. of)
136	Madagascar
137	Madeira (EU)
138	Malawi
139	Malaysia
140	Maldives
141	Mali
142	Malta
143	Marshall Islands
144	Mauritania
145	Mauritius
146	Mexico
147	Micronesia (Federated States)
148	Moldova
149	Monaco (EU)
150	Mongolia
151	Montenegro
152	Montserrat
153	Morocco
154	Mozambique
155	Myanmar (formerly Burma)

Country	
Country ID	Country Name
156	Namibia
157	Nauru Island
158	Nepal
159	Netherland Antilles
160	Netherlands (EU)
161	New Caledonia
162	New Zealand
163	New Zealand Island Territories
164	Nicaragua
165	Niger Republic
166	Nigeria
167	Norfolk Island
168	Northern Mariana Island
169	Norway
170	Oman
171	Pakistan
172	Panama
173	Papua New Guinea
174	Paraguay
175	Peru
176	Philippines
177	Pitcairn Island
178	Poland
179	Portugal (EU)
180	Puerto Rico
181	Qatar
182	Ras al Khaimah (use UAE)
183	Reunion
184	Romania
185	Russia
186	Rwanda
187	Sabah
188	Samoa (American)
189	San Marino (EU)
190	Sao Tome & Principe
191	Sarawak
192	Saudi Arabia
193	Senegal
194	Serbia
195	Seychelles
196	Sharjah (use UAE)
197	Sierra Leone
198	Singapore
199	Slovak Republic
200	Slovenia

Country	
Country ID	Country Name
201	Solomon Islands
202	Somalia
203	South Africa
204	South Georgia
205	South Sandwich Islands
206	Spain (EU)
207	Spanish North African Territories
208	Spitzbergen
209	Sri Lanka
210	St Christopher (St Kitts) & Nevis
211	St Helena
212	St Lucia
213	St Pierre & Miquelon
214	St Vincent & Grenadines
215	Sudan
216	Suriname
217	Swaziland
218	Switzerland
219	Sweden (EU)
220	Syria
221	Taiwan
222	Tajikistan
223	Tanzania
224	Thailand
225	Tibet
226	Togo
227	Tonga
228	Trinidad & Tobago
229	Tristan da Cunha
230	Tunisia
231	Turkey
232	Turkmenistan
233	Turks & Caicos Islands
234	Tuvalu
235	Uganda
236	Ukraine
237	Umm al Qaiwan (use UAE)
238	United Arab Emirates
239	Uruguay
240	USA
241	Uzbekistan
242	Vanuatu
243	Vatican City State

Country	
Country ID	Country Name
244	Venezuela
245	Vietnam
246	Virgin Islands (USA)
247	Wake Island
248	Wallis & Futuna Islands
249	Western Samoa
250	Yemen
251	Zambia
252	Zimbabwe

Appendix B, C, D, E

Please note to find the printing station ids, envelope id's and paper types you need to refer to the Letter Pricing API which can be found at the following link.

<http://www.pc2paper.co.uk/LetterPricingAPI.pdf>

The letter pricing API gives you a selection of available options based on the country you are sending your letter from. Calling the Letter Pricing API with your country code (Appendix A) will give you postage options, paper types, special options, printing types and envelopes types available for your letter.

You can read more about our API's over here

<http://www.pc2paper.co.uk/apis.asp>

