



**RedAPI
HTTP Reply Service
User Guide**

Version 1.00

12 April 2010

Red Oxygen Contact Details

Australia (Head Office)

Sales &
General Enquiries: +61 7 3100-8836
Accounts & Billing: +61 7 3100-8831
Technical Support +61 7 3100-8833
Fax: +61 7 3100-8838
Address: Red Oxygen Pty Ltd,
 Suite 39, Level 9,
 445 Upper Edward Street,
 Brisbane, QLD 4000

Postal Address: GPO Box 2452,
 Brisbane, QLD 4001

Sales: sales@redoxxygen.com
Support: support@redoxxygen.com

United States

Phone: +1 469-230-5018

Address: P.O. Box 41268
 Austin, Texas 78704

Sales: sales@redoxxygen.com
Support: support@redoxxygen.com

United Kingdom

Phone: +44 20 7987-2222
Address: 27 Old Gloucester Street,
 London WC1N 3AX

Sales: sales@redoxxygen.com
Support: support@redoxxygen.com

Revision Table

VERSION	PRIMARY AUTHOR(S)	DESCRIPTION OF VERSION	DATE COMPLETED
1.0	Tim Hunt	Initial Production Version documentation.	12 Apr 2010

Contents

Introduction	4
HTTP XML details	4
<i>XML Paramaters</i>	<i>4</i>
<i>XML Example:</i>	<i>4</i>
Client Web Server details	6
<i>Response from the Client Web Server</i>	<i>6</i>
<i>Sample Client Web Server (ASP) Code:.....</i>	<i>7</i>

Introduction

The Red Oxygen HTTP Reply Service provides a simple interface for programmers to receive SMS reply to their Web Server in XML format.

HTTP XML details

The Red Oxygen HTTP Reply Service will format the sms reply in the following XML format, and use Http Post the request to their Web Servers.

```
<?xml version="1.0" encoding="UTF-8"?>
<DATA>
  <CustomerId>[CustomerId]</CustomerId>
  <EmailId>[EmailId]</EmailId>
  <SmsId>[SmsId]</SmsId>
  <UserEmail>[UserEmail]</UserEmail>
  <UserName>[UserName]</UserName>
  <DestTN>[DestTN]</DestTN >
  <DestName>[DestName]</DestName>
  <ReplyText>[ReplyText]</ReplyText>
</DATA>
```

Where [...] is a place where the related value is stored.

XML Paramaters

Parameter	Description
CustomerId	NUMBER,
EmailId	CHAR (24), /* Numeric */
SmsId	CHAR (24), /* Numeric */
UserEmail	CHAR (64), /* String – system user's email address */
UserName	CHAR (256), /* String – system user's name*/
DestTN	CHAR (24), /* String – replying user's phone number */
DestName	CHAR (256), /* String – replying user's name */
ReplyText	CHAR (4000), /* String – reply text */

XML Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<DATA>
  <CustomerId>1</CustomerId>
  <EmailId>81001201120271</EmailId>
  <SmsId>19232595</SmsId>
```

```
<UserEmail>test@redoxxygen.com</UserEmail>
<UserName>Test+User</UserName>
<DestTN>61417156776</DestTN >
<DestName>John+Smith</DestName>
<ReplyText>This+is+my+test+reply+message</ReplyText>
</DATA>
```

Please Note: The XML values are URL encoded.

Client Web Server details

The Client Web Server needs to have something like an ASP page to receive and process the reply.

This page will receive the XML transmitted from the Red Oxygen server platform, and should be able to extract the data from the XML then respond appropriately:

Response from the Client Web Server

If the Client Server received the http request and successfully processed the xml, then it should reply the response with result:000

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<BODY>
<RESULT>000</RESULT>
</BODY>
</HTML>
```

Else if is error, It should return with the result. We do not store the result other than Delivered/Failed, but it can be seen in log files when investigating issues.

In the following example we take result = 100 to mean no xml block found:

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">
<HTML>
<BODY>
<RESULT>100</RESULT>
</BODY>
</HTML>
```

Sample Client Web Server (ASP) Code:

```
<%@ Language=VBScript %>
<%
    szPageTitle = "TestHttpReply"

    szPageUser = Request.QueryString("PageUser")
    szPagePass = Request.QueryString("PagePass")
    if szPageUser = "MyUser" and szPagePass = "MyPass" then
        szErrorCode = "000"
    else
        szErrorCode = "-10"
    end if

    szTestString = Request.QueryString("TestString")
    szRespText = Replace(Request.Form, "'", "'\"")

    if szRespText = "" then
        if szTestString = "Yes" then
            '/* This code segment is provided for testing purposes, it emulates */
            '/* the Red Oxygen calling this page by use of recursion, (i.e. it */
            '/* calls itself, with a valid reply data. */
            '/* To call this segment, you need to add the "&szTestString=Yes" */
            '/* to the URL. */
            '/* In this format, this code would never be called by if the reply */
            '/* is instigated by the Red Oxygen server because the szRespText */
            '/* would never be "", and is safe to leave here. */
            szRespText = "<?xml version=""1.0"" encoding=""UTF-8""?>" & vbCRLF _
                & "<DATA>" & vbCRLF _
                & " <CustomerId>1</CustomerId>" & vbCRLF _
                & " <EmailId>81001201120271</EmailId>" & vbCRLF _
                & " <SmsId>19232595</SmsId>" & vbCRLF _
                & " <UserEmail>test@redoxygen.com</UserEmail>" & vbCRLF _
                & " <UserName>Test+User</UserName>" & vbCRLF _
                & " <DestTN>0404444444</DestTN >" & vbCRLF _
                & " <DestName>Dest+User</DestName>" & vbCRLF _
                & " <ReplyText>This+is+my+test+reply</ReplyText>" & vbCRLF _
                & "</DATA>" & vbCRLF

            szURL = "Http://" & Request.servervariables("SERVER_NAME") & _
                Request.servervariables("PATH_INFO") & "?" & Request.QueryString

            set objHttp = Server.CreateObject("Msxml2.ServerXMLHTTP")
            objHttp.open "POST", szURL, false
            objHttp.setRequestHeader "Content-Type", "application/x-www-form-urlencoded"
            objHttp.Send szRespText

            if (objHttp.status <> 200 ) then
                szErrorCode = "200"
                szSkipToEnd = "No"
            else
                szResponse = objHttp.responseText
                szResult = Replace(szResponse, "<BODY>", "<BODY>Recursive call using test data<BR>")
                Response.write(szResult)
                szSkipToEnd = "Yes"
            end if
            set objHttp = nothing
        else
            szErrorCode = "100"
        end if
    end if

    if szSkipToEnd <> "Yes" then
        '/* This is the pain body of the program. First we log onto an Oracle Database. */
```

```

szSkipDB      = Request.QueryString("SkipDB")

if szSkipDB = "" then
    szDatabase      = "MY_ORA_DB"
    szDBUser        = "scott"
    szDBPass        = "tiger"

    Set DBConn = Server.CreateObject("ADODB.Connection")
    DBConn.ConnectionTimeout = 15
    DBConn.CommandTimeout = 15
    DBConn.CursorLocation = 3
    DBConn.Open "Provider=MSDAORA.1; Data Source=" & szDatabase & "; User Id=" & _
        szDBUser & "; Password=" & szDBPass & "; " & _
        "Persist Security Info=True; Max Pool Size=20; Min Pool Size=1", _
        szDBUser, szDBPass

    if DBConn.Errors.Count > 0 then
        szErrorCode = "-99"
    end if
end if

if InStr(szRespText, "<") > 0 then
    /* Here we split the XML up into Key - Value pairs, nd assign them to variables. */
    aReplyTmp = split(szRespText, "<")
    i = 0
    do while left(aReplyTmp(i), 6) <> "/DATA>"
        if InStr(aReplyTmp(i), ">") > 0 then
            aSplitTemp = split(aReplyTmp(i), ">")
            if aSplitTemp(0) = "CustomerId" then
                szCustomerId = aSplitTemp(1)
            elseif aSplitTemp(0) = "EmailId" then
                szEmailId = aSplitTemp(1)
            elseif aSplitTemp(0) = "SmsId" then
                szSmsId = aSplitTemp(1)
            elseif aSplitTemp(0) = "UserEmail" then
                szUserEmail = aSplitTemp(1)
            elseif aSplitTemp(0) = "UserName" then
                szUserName = aSplitTemp(1)
            elseif aSplitTemp(0) = "DestTN" then
                szDestTN = aSplitTemp(1)
            elseif aSplitTemp(0) = "DestName" then
                szDestName = aSplitTemp(1)
            elseif aSplitTemp(0) = "ReplyText" then
                szReplyText = aSplitTemp(1)
            end if
        end if
        i = i + 1
    loop
end if

/* And then insert the data into a table, for later processing... */
sql="INSERT INTO http_reply ( CustomerId, EmailId, SmsId, UserEmail, UserName, " & vbCRLF _
    & "                                DestTN, DestName, ReplyText, ErrorCode ) " & vbCRLF _
    & "VALUES ('" & szCustomerId & "', '" & szEmailId & "', '" & szSmsId & "', '" & vbCRLF _
    & "            '" & szUserEmail & "', '" & szUserName & "', '" & szDestTN & "', " & vbCRLF _
    & "            '" & szDestName & "', '" & szReplyText & "', '" & szErrorCode & "')"

if szSkipDB = "Yes" then
    /* For debug purposes we return the SQL from the above database insert statement */
    szResult = "Http://" & Request.servervariables("SERVER_NAME") & _
        Request.servervariables("PATH_INFO") & "<BR><PRE>" & _
        szRespText & vbCRLF & sql & "</PRE>"
else
    Set Newsrst = DBConn.Execute (sql)
    szResult = "<RESULT>" & szErrorCode & "</RESULT>"
end if

```

```
    '/* And Finally return the result. */
    Response.write("<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 3.2 Final//EN">" & vbCRLF)
    Response.write("<TITLE>" & szPageTitle & "</TITLE>" & vbCRLF)
    Response.write("<HTML>" & vbCRLF)
    Response.write("<BODY>" & vbCRLF)
    Response.write(szResult & vbCRLF)
    Response.write("</BODY>" & vbCRLF)
    Response.write("</HTML>" & vbCRLF)
end if
%>
```

This sample code could be called with the following URL:

<http://localhost/TestHttpReply.asp?PageUser=MyUser&PagePass=MyPass&TestString=Yes&SkipDB=Yes>

Where TestString=Yes and SkipDB=Yes are parameters passed for testing purposes only.