



Technical Interface v3.0

To test any of our interfaces, just obtain a free 24X username and password by going to www.24x.com/try Your account details will be sent to your mobile phone and can be used in the http, web service, email to SMS interfaces as well as the web and desktop applications – you will have 10 credits to test with and more can easily be purchased via credit / debit card by going to www.24x.com/buy

HTTP Interface

<http://www.24x.com/sendsms/sendsms.aspx>

Note: If you want to receive inbound replies back to an email address, then see the optional 'email' parameter below.

Required Fields

User	The account username assigned by 24x
Password	The account password
Smsto	The mobile number to send the message to. UK numbers do not require international coding. If sending the same message to multiple mobiles this field will accept up to 100 numbers delimited with commas.
Smsfrom	The originator of the message. Will accept 11 alphanumeric characters or longer if just numeric.
Smsmsg	The message content. This will vary dependent on the type of message being sent.

Optional Fields

Smstype	<p>1 = Ordinary Text Message</p> <p>2 = Nokia Ringtone ^</p> <p>3 = Nokia Group Logo *</p> <p>4 = Nokia Picture Message *</p> <p>5 = EMS Picture *</p> <p>6 = Nokia Operator Logo *</p> <p>7 = Flash Message</p> <p>8 = Nokia WAP Bookmark (See instruction below)</p> <p>9 = WAP Push (See instruction below)</p> <p>Default = 1</p> <p>^ Requires the smsmsg to in a valid Nokia RTTTL format</p> <p>* Requires the smsmsg to be hex encoded image data</p>
smsdate	Messages can be scheduled up to 12 months in advance. The format of

	the date is yyyy-mm-dd hh:MM:ss . If the date supplied is not in the valid format or cannot be evaluated to a valid date then the message(s) will be sent immediately.
Smsoperator	Required for Nokia Operator Logos 1 = O2 (BTCellnet) 2 = Vodafone 3 = TMobile (One2One , Virgin) 4 = Orange
Userfield	Custom field for your use up 50 characters.
Returnurl	You can specify a page to redirect to after sending the messages. This field will need to be a fully qualified URL e.g. http://www.24x.com
email	If an email address is passed in the posting, then the smsfrom field is replaced with a 24X response number. If the receiver replies to the text message, then 24X will route the reply back to the email address that was passed in the original posting to us. The smsfrom must still be posted to us.
long	If you pass the parameter long=Y then your message length can exceed 160 characters enabling long messages to be sent. Once you go over 160 the breakpoints are at 153 characters as the remaining spaces are required to join the message so credits are taken per 153 character length or part thereof ie. 306 = 2 credits (2 x 153).
Batchref	If you want us to 'throttle' the sending of messages over a period of time then pass us a batch reference value that is the same value in each posting that constitutes a batch. This is used in conjunction with our batch management system to define the time period, cancel or speed up batches. Length = 25 (no spaces).

Responses

00 - n	Success (n is the 24x assigned ID for the message. A comma delimited list of IDs will be sent if the smsto contained more than one mobile number.)
01	Username or password incorrect.
02	smsTo missing.
03	Smsfrom missing.
04	Smsmsg missing.
05	Smstype invalid.
06	Send Date greater that one month in the future.
07	Invalid operator code.
08	Operator required for Nokia logo.
09	Insufficient credits.
10	More than 100 mobile numbers in then smsTo field
11	WAP Push combined field length greater than 90 characters

The following is an example of sending a standard message to one phone.

<http://www.24x.com/sendsms/sendsms.aspx?user=xxxxx&password=xxxxxx&smsto=441112223334&smsfrom=24x&smsmsg=24xTest&smstype=1>

The following is an example of sending a standard message to multiple phones.

<http://www.24x.com/sendsms/sendsms.aspx?user=xxxxx&password=xxxxxx&smsto=441112223334.443334445556,446667778889&smsfrom=24x&smsmsg=24xTest&smstype=1>

Send a Nokia Wap Bookmark

User	The account username assigned by 24x
Password	The account password

smsto	The mobile number to send the message to. UK numbers do not require international coding. If sending the same message to multiple mobiles this field will accept up to 100 numbers delimited with commas.
Smsfrom	The originator of the message. Will accept 11 alphanumeric characters or longer if just numeric.
smstype	Must always be 8
Bookmarktype	Must always be NOKIA in UPPERCASE
BookmarkName	The name that will appear in the list of bookmarks
Bookmarkurl	The full URL of the WAP site.

Picture Messages - Send a Wap Push (MMS message with URL)

User	The account username assigned by 24x
Password	The account password
smsto	The mobile number to send the message to. UK numbers do not require international coding. If sending the same message to multiple mobiles this field will accept up to 100 numbers delimited with commas.
Smsfrom	The originator of the message. Will accept 11 alphanumeric characters or longer if just numeric.
smstype	Must always be 9
smsmsg	This field will be the URL of the site to navigate to.

Example

<http://www.24x.com/sendsms/sendsms.aspx?user=xxxx&password=xxxx&smsmsg=http://www.abc.com/xyz.jpg&smsto=447712123256&smsfrom=24x&smstype=9>

Administration options are available through the following syntax:-

<http://www.24x.com/sendsms/smsadmin.aspx>

Required Fields

User	The account username assigned by 24x
Password	The account password
Service	Valid entries Credits = Account balance. Time = 24x server time to allow for accurate scheduling. Deletequeued = Removes a message from the scheduled queue (requires the 24x unique message id) Reschedule = Change the sending date and time of a scheduled message (requires the 24x unique message id and a new date and time)

e.g. <http://www.24x.com/sendsms/SMSADMIN.aspx?user=xxx&password=xxxx&service=credits>

Optional Fields

id	The unique 24x message id
date	The new date and time for a scheduled message. Required when using the reschedule service

Responses

00 - n	Success for delete queued and reschedule services (n is the number of messages affected by the action).
01	Username or password incorrect.
02	Message ID Invalid
03	No scheduled messages to delete.
04	Send date > 1 month in advance.
05	Date not valid.
06	No scheduled messages to reschedule.

Delivery Reports

If you create a page or script and email us the URL then we will post delivery reports to you via an HTTP POST. (email kevin.strong@24x.com with the URL)

The fields passed are

id
status
statustime in the format DD/MM/YYYY HH:MM:SS (space between date and time).

The id is the 24x unique id given to you when you posted the message to us.

The status will be either DELIVERED or FAILED

The statustime is the time that the message was delivered or failed.

Note: Messages usually deliver in approximately 7 seconds BUT delivery reports may take from seconds to days to be passed back to us – This is not a reflection of the message delivery time, just the low priority that operators give to delivery reports.

If you want to start sending messages quickly just "Cut and Paste" our example code into an ASP or VB application and use the SendMessage function. (Web Service, PHP and Cold Fusion script sample are at the end of this document) Note: we use charset=iso-8859-1 because it supports £ signs:-

```
-----  
result = SendMessage ("24Username", "24Xpassword", "07123456789", "24xLtd", "Hello World")
```

```
Function SendMessage(sUserName, sPassword, sMobileNumber, sSenderNumberOrName, sMessage)
```

```
Dim xmlhttp, sResponse, sUN, sPW, sTO, sFR, sMESS
```

```
sUN = URLEncode(sUserName)  
sPW = URLEncode(sPassword)  
sTO = URLEncode(sMobileNumber)  
sFR = URLEncode(sSenderNumberOrName)  
sMESS = URLEncode(sMessage)  
Set xmlhttp = CreateObject("MSXML2.ServerXMLHTTP")
```

```
xmlhttp.Open "POST", "http://www.24x.com/sendsms/sendsms.aspx", False
```

```
xmlhttp.setRequestHeader "Content-Type", "application/x-www-form-urlencoded; charset=iso-8859-1"
```

```
xmlhttp.send "user=" & sUN & "&password=" & sPW & "&smsto=" & sTO & "&smsfrom=" & sFR & "&smsmsg=" & sMESS
```

```
sResponse = xmlhttp.responseText
```

```
Set xmlhttp = Nothing
```

```
SendMessage = sResponse
```

```
End Function
```

Function URLEncode(strData)

Dim I, strTemp, strChar, strOut, intAsc

strTemp = Trim(strData)

For I = 1 To Len(strTemp)

strChar = Mid(strTemp, I, 1)

intAsc = Asc(strChar)

If (intAsc >= 48 And intAsc <= 57) Or _
(intAsc >= 97 And intAsc <= 122) Or _
(intAsc >= 65 And intAsc <= 90) Then
strOut = strOut & strChar

Else

strOut = strOut & "%" & Hex(intAsc)

End If

Next

URLEncode = strOut

End Function

VB.Net code example -

```
Imports System.Net
Imports System.IO
Imports System.Web

Function SendSMS(ByVal strUserName As String, ByVal strPassword As String,
ByVal strSMSTo As String, ByVal strSMSFrom As String, ByVal strSMSMsg As
String) As String

Dim objResponse As WebResponse
Dim objRequest As WebRequest
Dim strURL As String
Dim strData As String
Dim strResult As String

strURL = "http://www.24x.com/sendsms/sendsms.aspx?"

strSMSMsg = HttpUtility.UrlEncode(strSMSMsg)

strSMSFrom = HttpUtility.UrlEncode(strSMSFrom)

strData = "user=" & strUserName & "&password=" & strPassword & "&smsto=" &
strSMSTo

strData &= "&smsfrom=" & strSMSFrom & "&smsmsg=" & strSMSMsg

strURL = strURL & strData

objRequest = HttpWebRequest.Create(strURL)

objResponse = objRequest.GetResponse()

Dim objSR As New StreamReader(objResponse.GetResponseStream())

strResult = objSR.ReadToEnd().TrimEnd

Return strResult

End Function
```

Email To SMS Interface

Individual emails – These are sent to an email address in the following format:-
The **To** address should mobilenumber.uniqueID@24xgateway.com

Please contact us to receive your unique ID.

There are no setup charges, you only pay for message credits as per a normal SMS account.

Email To SMS Interface – with replies

This system enables you to send an SMS via email and if the user responds, the reply is received back in your email inbox.

This system costs £20 mth plus your normal message credits cost.

Inbound – (Receiving SMS Messages from a long number)

In order to receive inbound messages you need to set up a page or script on your server to receive four parameters (Number,Message,date,id) in real time as they are received by us. The id is a unique message number that you should check to ensure that you are not receiving duplicate messages. Your page or script should return OK back to our system when we POST to it. Please refer to the variables by name (**id, Number, Message, date**) NOT by their position in the string.

The requirements are as follows:-

```
http://www.youURL.com/inbound.asp?id=123456&Number=4455667788&Message=Hello_World&date=11/06/2002 10:29:30
```

Date format is DD/MM/YYYY HH:MM:SS

Asp.net example to use to display data that we post to you:-

```
Protected Sub Page_Load(ByVal sender As Object, ByVal e As System.EventArgs) Handles Me.Load
```

```
Dim iID As Integer = Request.Params("id")
Dim strMessage As String = Request.Params("message")
Dim strNumber As String = Request.Params("number")
Dim dDate As Date = Request.Params("date")
```

```
Dim strReceived As String
```

```
strReceived = "ID = " & iID.ToString & "<br/>"
```

```
strReceived &= "Message = " & strMessage & "<br/>"
```

```
strReceived &= "Mobile = " & strNumber & "<br/>"
```

```
strReceived &= "Date = " & dDate.ToString
```

```
Response.Write(strReceived)
```

```
Response.End()
```

```
End Sub
```

Inbound – (Receiving messages sent to a short code)

In order to receive inbound messages you need to set up a page or script on your server to receive six parameters (**id, mobile, shortcode, message, network, date, keyword**) in real time as they are received by us. We will **POST** the variables to your page so you need to extract the variables using request.form NOT request.querystring

The requirements are as follows:-

```
http://www.youURL.com/inbound.asp?id=35558965&mobile=4455667788&shortcode=88222&message=Hello_World&shortcode=87222&network=BTCELLNETUK&id=1122334455&date=11/6/2002 10:29:30&keyword=TEST
```

The possible values for network are BTCELLNETUK , VODAFONEUK , ORANGEUK , TMOBILEUK and 6 (6 is the 3 Network).

You must send an OK back to our system to acknowledge receipt of the posting. Also, you must check the id that we send you as it is unique. Ignore any repeat postings with the same id as this will be a duplicate however we STILL require an OK returned to us.

In order to setup a reverse billing system:-

You need to decided which tariff you will be using (25p, 50p, £1.00, £1.50) We will allocate a keyword to be used for inbound SMS.

You setup a script / page on your server that will accept 6 parameters – mobile number, date/time, message, ID, shortcode and network - we will post inbound SMS to this page in 'real time'.

If you are subscribing people through the web then you need to capture their mobile number and network.

We allocated you a username and password and you must bill the particular mobile number using the same http posting interface that is at the front of this document. The ONLY thing you need to do is to use a different username and password to the one used for standard messages.

We also allocate you a standard account for which you purchase standard credits in advance for delivery of the content.

Use the billing accounts for billing messages only and the standard account to deliver the content.

Receiving MMS messages inbound

Inbound MMS messages are delivered to an ftp area on your server. The following files could be transferred – they will ALL have the same filename with different extensions:-

Filename.txt - This is the message text within the message
Filename.jpg - This is a picture
Filename.3gpp - This is a 3 network video file

The format of the filename is mobilenumber_ddmmyyyy_hhmmss_uniquenumber
Please note that it does not have the extension type on it.

In order to receive inbound messages you need to set up both an ftp receiving area and tell us the username and password. At the end of the ftp transfer process we can call a page or script on your server which can then action any processing you require. Please let us know the URL to call.

SMS sending - Web Service

A web service is also available if required, please call us for details.

A short example of Posting via PHP.

```
<?php
$host = "www.24x.com";
$path = "/sendsms/sendsms.aspx";
$username = "xxxxxx";
$password = "xxxxxx";
$smsto = "447777777777";
$smsfrom = "24X";
$smsmsg = "testing " . date("d/m/Y H:i:s");

$data = 'user=' . urlencode($username);
$data .= '&password=' . urlencode($password);
$data .= '&smsto=' . urlencode($smsto);
$data .= '&smsfrom=' . urlencode($smsfrom);
$data .= '&smsmsg=' . urlencode($smsmsg);
```

```
echo sendTo24X($host,$path,$data);

function sendTo24X($host,$path,$data)
{

    $fp = fsockopen($host, 80);

    $header = "POST $path HTTP/1.1\r\n";
    $header .= "Host: $host\r\n";
    $header .= "Content-type: application/x-www-form-urlencoded; charset=iso-8859-1\r\n";
    $header .= "Content-length: " . strlen($data) . "\r\n";
    $header .= "Connection: close\r\n\r\n";

    fputs($fp,$header);

    fputs($fp,$data);

    while (!feof($fp))
    {
        $buf .= fgets($fp,1024);
    }

    fclose($fp);

    return $buf;
}

?>
```

PHP sample sending code with form. Code courtesy of Martin Hope MXBmedia

24x Username:

24x Password:

To Mobile Number:

From Name:

Message Text:

```
<?php
    // 24xtest1.php - PHP Script to test basic http requests to 24x SMS Gateway.

$username = "";
$password = "";
```

```

#
*****
*****
# Process a submitted form
#
*****
*****
if(isset($_POST['submit'])) {
    $username = urlencode($_POST['username']);
    $password = urlencode($_POST['password']);
    $params = "user=$username&password=$password";
    $msg = "24x Response: ";

    switch($_POST['submit']) {
        #
        *****
        *****
        # Send SMS
        case "Send SMS":
            # Get form input
            $smsTo = urlencode($_POST['smsTo']);
            $smsFrom = urlencode($_POST['smsFrom']);
            $smsMsg = urlencode($_POST['smsMsg']);
            $params .= "&smsto=$smsTo&smsfrom=$smsFrom&smsmsg=$smsMsg";
            $script = "/sendsms/sendsms.aspx";
            break;

        #
        *****
        *****
        # Check Credit
        case "Check Credit":
            $params .= "&service=credits";
            $script = "/sendsms/smsadmin.aspx";
            break;

        #
        *****
        *****
        # Show Code
        case "View PHP Code":
            header("Location: 24xtest1.txt");
            break;

        #
        *****
        *****
        default:
            // Do nothing
            break;
    }

    #
    *****
    *****
    # send the request
    $HOST = "www.24x.com";
    $logNo= 0;

    $header = "POST $script HTTP/1.0\r\n";
    $header .= "Host: $HOST\r\n";
}

```

```

$header .= "Content-Type: application/x-www-form-urlencoded\r\n";
$header .= "Content-Length: ".strlen($params)." \r\n\r\n";

$fp = @fsockopen ($HOST, 80, $logNo, $errstr, 30);

if (!$fp) {
    // HTTP ERROR
    $msg = "HTTP ERROR: FAILED to connect to $HOST";
} else {
    fputs ($fp, $header.$params);
    while (!feof($fp)) {
        $res = fgets ($fp, 1024);
    }
    $msg .= $res;
}
@fclose ($fp);
echo "$msg<br>";

}
#
*****
*****
#
*****
*****

?>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Send SMS via 24x PHP Test</title>
</head>

<body>
<form name="sendRequest" id="sendRequest" method="post"
action="24xtest1.php">
<table border="0" cellpadding="0">
    <tr>
        <td width="160" height="30" align="left">
            24x Username:
        </td>
        <td align="left">
            <input type="text" name="username" id="username"
size="16" value="<?php echo $username; ?>" />
        </td>
    </tr>

    <tr>
        <td width="160" height="30" align="left">
            24x Password:
        </td>
        <td align="left">
            <input type="text" name="password" id="password"
size="16" value="<?php echo $password; ?>" />
        </td>
    </tr>

```



```
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1" />
<title>Send SMS</title>
```

```
<script type="text/javascript" language="javascript">
```

```
    var username = "*****"; //24x username
    var password = "*****"; //24x password
```

```
// Standard HTTP request check to determine browser - DO NOT EDIT
```

```
var http_request = false;
```

```
function makePOSTRequest(url, parameters) {
```

```
    http_request = false;
```

```
    if (window.XMLHttpRequest) { // Mozilla, Safari,...
```

```
        http_request = new XMLHttpRequest();
```

```
        if (http_request.overrideMimeType) {
```

```
            http_request.overrideMimeType('text/html');
```

```
        }
```

```
    } else if (window.ActiveXObject) { // IE
```

```
        try {
```

```
            http_request = new ActiveXObject("Msxml2.XMLHTTP");
```

```
        } catch (e) {
```

```
            try {
```

```
                http_request = new ActiveXObject("Microsoft.XMLHTTP");
```

```
            } catch (e) {}
```

```
        }
```

```
    }
```

```
    if (!http_request) {
```

```
        alert('Cannot create XMLHTTP instance');
```

```
        return false;
```

```
    }
```

```
// end standard browser check
```

```
    // start http post request
```

```
    http_request.onreadystatechange = alertContents;
```

```
    http_request.open("POST", "http://www.24x.com/sendsms/sendsms.aspx", false);
```

```
    http_request.setRequestHeader("Content-type", "application/x-www-form-urlencoded");
```

```
    http_request.send(parameters);
```

```
}
```

```
function alertContents() {
```

```
    if (http_request.readyState == 4) {
```

```
        if (http_request.status == 200) {
```

```
            result = http_request.responseText;
```

```
        } else {
```

```
            alert('There was a problem with the request.');
```

```
        }
```

```
    }
```

```
}
```

```
function get(obj) {
```

```
    var smsto = encodeURIComponent( document.getElementById("TextBox1").value ); //recipients phone number
```

```
    var smsfrom = encodeURIComponent( document.getElementById("TextBox2").value ); //senders name
```

```
    var smsmsg = encodeURIComponent( document.getElementById("TextBox3").value ) //text message content
```

```
    var poststr = "user=" + username + "&password=" + password + "&smsto=" + smsto + "&smsfrom=" + smsfrom + "&smsmsg=" + smsmsg;
```

```
    makePOSTRequest('http://www.24x.com/sendsms/sendsms.aspx', poststr);
```

```

}
</script>
</head>
<body>
    <div id="MainContainer">
        <div id="PageHead"></div>
        <div id="PageBody">

            <form action="javascript:get(document.getElementById('myform'));" name="myform" id="myform">

                <h1><span style="font-family: Arial">Send SMS</span></h1>
                <p>
                    To: <input type="text" id="TextBox1" name="TextBox1" />
                </p>
                <p>
                    From: <input type="text" id="TextBox2" name="TextBox2" />
                </p>
                <p>
                    <span style="font-family: Arial">Message:</span></p>
                <p>
                    <input type="text" id="TextBox3" name="TextBox3" />
                </p>
                <p>
                    <input type="submit" name="button" value="Send Text Msg">

                </p>
            </form>

        </div>
        <div id="PageFoot"></div>
    </div>
</body>
</html>

```

PHP sample code to send messages via email to SMS :-

This will send an SMS if the user is signed up for it, and in credit, and an email if not.

This script is called mailer.inc and is called when an email/SMS is required

PHP:

```

<?php
# This code is called with message, receiver, and subject set in variables
# Then it sends an SMS or email as per the settings in php.ini, from the superuser as set in the
config.inc.php file

# Note the use of \n for newline in the headers. Some MTAs accept this fine, others will need \r\n
instead of just \n
# Some MTAs will put the \r in for you and the headers won't work, so try and see which works best
for you and your MTA
# If you've got the wrong combination, then the html won't get rendered and/or other header problems
will arise
if (($adminmessageinstead=='y') and ($email_recipientname==$superuser)) {
    # Send as an internal private message to sysadmin instead of SMTP
    mysql_query("insert into messages values
('',now(), 'System', '$superuser', '$email_subject', '$email_message', 'n', 'n')") or die(mysql_error());
} else {
    # Send an SMS or SMTP email instead
    # Prevent guest login from sending emails or texts

```

```

if (!(($innow=='Guest') or ($innow=='Guest Instructor'))) {
    # See if we can send an SMS instead
    $sendingemail='y';
    if ($email_smspreferred=='y') {
        # See if they've opted in for it and got credits and a mobile number
        $sms_valid=mysql_fetch_row(mysql_query("select
mobilenumber,smsregistered,smscredit from people where ((username='$email_recipientname') and
(mobilenumber is not null)"));
        if ($sms_valid) {
            # Send an SMS
            if (($sms_valid[0] and $email_smsanyway=='y') or ($sms_valid[0] and
$sms_valid[1]=='y' and $sms_valid[2]>0)) {
                $headers = "MIME-Version: 1.0\n";
                $headers .= "Content-Type: text; charset=ISO-8859-1\n";
                $headers .= "From: ".$superuser." <".$superuseremail.">\n";
                $headers .= "X-Mailer: PHP mail $schoolname";
                # Rewrite the email address to go to their mobile
                $email_recipientaddress=$sms_valid[0].$smsaccount;
                if (!($email_recipientaddress=='<unknown>')) mail($email_recipientaddress,
$email_subject, $email_message_sms, $headers);
                # Reduce their SMS credit
                mysql_query("update people set smscredit=smscredit-1 where
username='$email_recipientname'");
                # Tell system not to send an email
                # $sendingemail='n';
            }
        }
    }
}
# End
if (($email_recipientaddress) and ($sendingemail=='y')) {
    $headers = "MIME-Version: 1.0\n";
    $headers .= "Content-Type: text/html; charset=ISO-8859-1\n";
    $headers .= "From: ".$superuser." <".$superuseremail.">\n";
    $headers .= "X-Mailer: PHP mail $schoolname";
    if (!($email_recipientaddress=='<unknown>')) mail($email_recipientaddress,
$email_subject, "<html><body>".$email_message, $headers);
} else {
    echo("<p><b>No valid email address at this time, note: Guest users aren't entitled to
the email facilities</b><p>");
};
} else {
    mail($superuseremail, $email_subject, "<html><body>".$email_message, $headers);
    # echo("<p><b>As you are a Guest user, an email has been sent to $superuser. But when
you have a valid account of your own the email will be sent to you and your instructor.</b><p>");
};
};
?>

```

Cold Fusion sample code:-

```

<cfif isdefined("form.fieldnames")>
    <cfset MobileToSMS = Replace(form.MobileToSMS," ","", "all")>
    <cfhttp url="http://www.24x.com/sendsms/sendsms.aspx" method="post"
resolveurl="no">
        <cfhttpparam type="formfield" name="user" value="#form.userId#">

        <cfhttpparam type="formfield" name="password"
value="#form.password#">
        <cfhttpparam type="formfield" name="smsto"
value="#form.MobileToSMS#">

```

```

        <cfhttpparam type="formfield" name="smsfrom"
value="#form.from#">
        <cfhttpparam type="formfield" name="smsmsg"
value="#Form.SMSMessage#">
    </cfhttp>
</cfif>

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
<html>
<head>
<meta http-equiv="Content-Type" content="text/html; charset=iso-8859-1">
<title>SMS Sender</title>
</head>

<body>
<form name="form1" method="post" action="">
  <table width="100%" border="0" cellpadding="5">
    <tr>
      <td>User Id </td>
      <td><input name="UserId" type="text" id="UserId"></td>
    </tr>
    <tr>
      <td>Password</td>
      <td><input name="Password" type="text" id="Password"></td>
    </tr>
    <tr>
      <td>From</td>
      <td><input name="from" type="text" id="from"></td>
    </tr>
    <tr>
      <td>Number</td>
      <td><input name="MobileToSMS" type="text" id="MobileToSMS"></td>
    </tr>
    <tr>
      <td>Message</td>
      <td><input name="SMSMessage" type="text" id="SMSMessage" size="160"
maxlength="160"></td>
    </tr>
    <tr>
      <td><input type="submit" name="Submit" value="Submit"></td>
      <td>&nbsp;</td>
    </tr>
  </table>
</form>

<cfif isdefined("form.fieldnames")>
<h1>Message Sent</h1>
</cfif>

</body>
</html>

```

If you need any more information or have any queries, please contact 24X on 01903 228100 or email kevin.strong@24x.com