



Connecting to 2sms

About 2sms

2sms allows business customers to send text messages from your computer to cell phones anywhere in the world. We help customers improve operations, demonstrate duty of care, and react more responsively to customers.

Text messaging reaches more people than email and faster than calling.

- ✓ To learn more - www.2sms.com
- ✓ Want a No cost No obligation trial account? - www.2sms.com/register
- ✓ For technical information - www.2sms.com/software



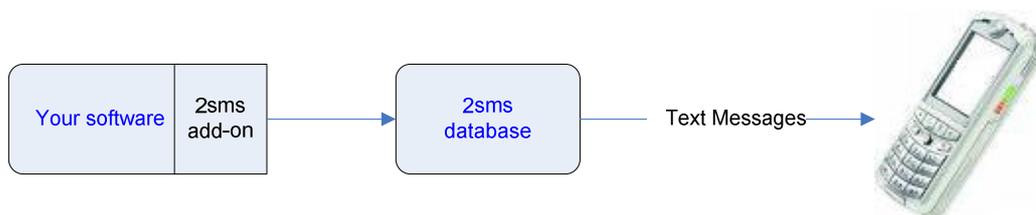
For Help, call 2sms on 877-276-7266 (USA), 0800 65 222 77 (UK) or +44 1234 757 800 (Rest of World) or email support@2sms.com. Support is available 24/7.

Connecting to 2sms Text Messaging Systems

This section explains how customers connect into 2sms systems for the purpose of sending a text message.

3 parts

1. Customer software choice for text message sending.
2. Establish a connection between the customer and 2sms
3. How 2sms delivers text messages.



Customer Doc #	SAL-190
Issue No./Date	1.0 2-Mar-07
Classification	N/A

Customer Software choice for Text Message Sending

Customers use a number of software solutions provided by 2sms to send text messages. Details of these options are available at www.2sms.com/software.

Depending on the customers choice, some software may need to be installed inside the customer's firewall. If the 2sms web site is used to send text messages, then access to www.2sms.com will be required on the customer's proxy server, if a proxy server is installed.

If any of the other 2sms software solutions are used, with the exception of email to SMS and SNPP, then a connection to 2sms will need to be established

If the customer is installing 2sms software, then the IT administrator will need to manage and oversee installation of the 2sms solution chosen. Most 2sms software has been independently validated or certified by the relevant application vendor. For example, the 2sms IBM WebSphere portlet has been validated by IBM. Most 2sms software is also digitally signed using Verisign.

If any of the other 2sms software solutions are used, with the exception of email to SMS and SNPP, the method used to send a customers text messages from the customers network to 2sms is by sending XML over http to the 2sms servers. The 2sms servers then respond with XML which is sent back to the 2sms software, and is used by the 2sms software to confirm the message sending transaction, providing feedback on the success or failure of the transaction.

Establish a Connection between the Customer & 2sms

Option 1

Open a connection on HTTP port 80 to 2sms at ip addresses 66.117.39.254 (production site) and 62.232.20.2 (contingency site)

Option 2

Open a connection on HTTPS port 443 to 2sms at ip addresses 66.117.39.254 (production site) and 62.232.20.2 (contingency site)

Option 3

Register customers ip address on 2sms firewall, to only allow text messaging transactions for the customer

Option 4

Configure a reverse proxy setup to allow customer to confirm that the 2sms ip address is valid.

How 2sms Delivers Text Messages

On receipt of the text messaging data from a customers site, 2sms processes the data, and inserts it into a secure database table held on 2sms servers. From this database table, the text message is submitted into the text messaging network using 2sms's secure connections to mobile phone operators and/or text messaging aggregators (wholesalers). The text message is send on to the recipient handset, and confirmations of message transmission are passed back to 2sms, and the database tables are updated. Later on, delivery receipts, if provided by the destination network, are also passed back to 2sms, and the database tables are updated.

2sms will confirm message delivery status back to the customer, communicating with the software solution that the customer has used to send the text messages.

Additionally, the customer can access a range of resources to get reports on messages sent.

2sms is legally required to hold text messaging data for 2 years, and has elected to keep the data for seven years.

Connecting to Third Party Systems

This document explains how 2sms text messaging services can interact with

- E-Commerce sites
- Remedy
- SMTP

E-Commerce Sites

E-commerce sites use a range of different technologies, and run on a number of internet server / portal server technologies.

2sms supports all the leading server technologies

- Apache (Java)
- Microsoft IIS
- IBM WebSphere
- BEA WebLogic
- Oracle Portal server

For more information on how 2sms allows customers to connect e-commerce sites to text messaging, please read the 2sms software document at :

www.2sms.com/documents/US/Software/2sms_US_Software_Overview.pdf

Remedy

Connection to Remedy can be done using one of two methods.

- SMTP
- Web services

SMTP

Please refer to the SMTP section below.

Web Services

Remedy offers its customers the ability to connect into external third party services by making calls to 'Web Services'. 2sms provides access to its text messaging service using web services. Information on this protocol can be found at

www.2sms.com/software. Documentation explaining how web services provided by 2sms can be read by downloading the following document :
www.2sms.com/documents/US/Software/2sms_US_Software_Web_Services_Overview.pdf

Customer Doc #	SAL-190
Issue No./Date	1.0 2-Mar-07
Classification	N/A

SMTP

Overview

To connect using SMTP, Remedy needs to be configured to send an email in the format required by the 2sms email to SMS service, and then posted to the 2sms SMTP server by addressing email@2sms.com

Example email

An example of the email format is :

```
To : email@2sms.com
Subject : <leave blank>
Body:
id:xxx
password:yyy
message:<enter your text message here>
17036069996
17036069997
17036069998
17036069999
end
```

The spec for this email format is listed at :

www.2sms.com/documents/US/Software/2sms_US_Software_Email2sms_Overview.pdf

Example of How to Send Text Messages using SMTP

The following is an example of how to use programming code to to send a text messages using SMTP. The code is written in VBScript, which extracts mobile phone numbers from a SQL server database. This code can be modified to extract data from any data source. This code uses a third party SMTP object supplied by Easymail. Customers can substitute this with their own SMTP object.

```
<%
'
' Setup SMTP session using Easymail SMTP mail object
' available from www.easymail.com
'

SET SMTP = CreateObject("EasyMail.SMTP.6")

'
' enter your easymail license key
'

SMTP.LicenseKey = "xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx"

'
' specify your email server
'

SMTP.MailServer = "smtp.emailserver.com"

'
' Enter the text message you wish to send
'
```

Customer Doc #	SAL-190
Issue No./Date	1.0 2-Mar-07
Classification	N/A

```
SMSMsg = "<please enter your text message here>"
```

```
'  
' Setup your 2sms id and password  
'
```

```
2smsID = <enter your 2sms ID here>  
2smsPassword = <enter your 2sms password here>
```

```
'  
' form up the main body of the email message to be sent via SMTP  
'
```

```
Message = "id:" & 2smsID & chr(13)  
Message = Message & "password:" & 2smsPassword & chr(13)  
Message = Message & "message:" & chr(13)  
Message = Message & SMSMsg & chr(13)
```

```
'  
' Setup the email parameters  
'
```

```
SMTP.Subject = ""  
SMTP.From = "Joe Doe"  
SMTP.FromAddr = joe.doe@mycompany.com
```

```
'  
' run a database query to get cell phone numbers out of a database  
'
```

```
set stafflist = connect.execute("EXEC stafflistcredits")
```

```
do while not stafflist.eof
```

```
    Message = Message + rtrim(stafflist("cellphone")) + chr(13)  
    stafflist.movenext
```

```
loop
```

```
'  
' put in final line of email  
'
```

```
Message = Message + "end" + chr(13)
```

```
'  
' transmit the message over SMTP  
'
```

```
SMTP.BodyText = message  
x = SMTP.Send
```

```
'  
' close out the send message object  
'
```

Customer Doc #	SAL-190
Issue No./Date	1.0 2-Mar-07
Classification	N/A

Set SMTP = nothing

%>



Customer Doc #	SAL-190
Issue No./Date	1.0 2-Mar-07
Classification	N/A