



# **IntelliSoftware**

## **SMS Gateway - COM Component Interface**

Version 1.5

Content subject to change

IntelliSoftware Ltd.  
Arena Business Centre  
Holyrood Close  
Poole  
Dorset  
ENGLAND  
BH17 7FJ  
Tel: +44 (0)1202 311572  
Fax: +44 (0)870 762 0876  
Web: [www.intellisoftware.co.uk](http://www.intellisoftware.co.uk)

# Contents

- 1. Document History**
- 2. COM Component Interface Overview**
- 3. Submitting to our gateway**
  - 3.1. Send Text Message**
  - 3.2. Send Text Message to Multiple Recipients**
  - 3.3. Send WAP Push Message**
  - 3.4. Send Binary Message**
  - 3.5. Send Unicode Message**
  - 3.6. Get Credit Balance**
- 4. Collections**
  - 4.1. SendStatusCollection**
- 5. Enumerations**
  - 5.1. ResultCodes Enumeration**
  - 5.2. SendStatus Enumeration**
- 6. Advanced Features**
  - 6.1. Send to Distribution Lists (Group Send)**
  - 6.2. Reply Tracking**

## 1. Document History

Version	Date	Author	Description
1.0	09/12/2000	Chris Vine	First Release
1.1	25/03/2005	Chris Vine	Added Binary and WAP Push SMS Submissions
1.2	25/03/2005	Chris Vine	Added MaxConCatMsgs Property
1.3	19/10/2006	Chris Vine	Added Unicode Support and Advanced Features
1.4	09/05/2008	Chris Vine	Corrected error in ResultCodes Enumeration
1.5	27/04/2009	Chris Vine	Updated Company Address

## 2. COM Component Interface Overview

Using the IntelliSMS COM Component you can send SMS messages from any language that supports COM. This includes VB, VBScript, JScript, ASP, C++. This component is free to download and use.

### Methods:

<b>SendMessage</b>	Sends a standard SMS text message.
<b>SendMessageToMultipleRecipients</b>	Sends a standard SMS text message to multiple recipients.
<b>SendWapPushMessage</b>	Sends a WAP PUSH message. WAP PUSH messages can be used to send multimedia content to mobile phones.
<b>SendBinaryMessage</b>	Sends a Binary Message. Binary Messages can be used to send SmartMessages, such as Ringtones and Logos.
<b>SendUnicodeMessage</b>	Sends a Unicode Message.
<b>GetBalance</b>	Obtains the available credits on an account

### Properties:

Username	This is the username of your account
Password	This is the password for your account
MaxConCatMsgs	Maximum number of concatenated SMS messages that will be sent per recipient (Default is 1)
ProxyAddress	(Optional) If a proxy server is required to access the internet then set this property to the address of your proxy server

### Secure SSL Connection (HTTPS):

To use a secure internet connection configure the gateway addresses as follows:

```
<%  
IntelliSMS.PrimaryGateway = "https://www.intellisoftware.co.uk"  
IntelliSMS.BackupGateway = "https://www.intellisoftware2.co.uk"  
  
SendStatus = IntelliSMS.SendMessage ( .....  
%>
```

## 3. Submitting to our gateway

### 3.1. Send Text Message

**SendMessage ( To As String, Text As String, From As String, MessageId As Variant ) As ResultCodes**

#### Description:

Sends an SMS message via the Internet Gateway.

#### Parameters:

To	Input	This is the destination phone number
Text	Input	The content of the text message
From	Input	This is the source/sender's phone number
MessageId	Output	Unique message ID
ResultCode	Return	See <a href="#">ResultCodes</a>

#### ASP Example:

```
<%  
Set IntelliSMS = Server.CreateObject("IntelliSoftware.IntelliSMS")  
  
IntelliSMS.Username = "MyUsername"  
IntelliSMS.Password = "MyPassword"  
  
Result = IntelliSMS.SendMessage _  
    ( "44771012345", "Hello", "SENDER_ID", MessageId )  
%>
```

#### VB Example:

```
Dim IntelliSMS As New INTELLISMSLib.IntelliSMS  
  
IntelliSMS.Username = "MyUsername"  
IntelliSMS.Password = "MyPassword"  
  
Dim MessageId As Variant  
Dim Result As INTELLISMSLib.ResultCodes  
  
Result = IntelliSMS.SendMessage _  
    ( "44771012345", "Hello", "SENDER_ID", MessageId )
```

## 3.2. Send Text Message to Multiple Recipients

**SendMessageToMultipleRecipients ( To As String, Text As String, From As String ) As SendStatusCollection**

### Description:

Sends an SMS message to multiple recipients via the Internet Gateway.

### Parameters:

To	Input	This is the destination phone number(s) NOTE: Separate multiple numbers by a comma
Text	Input	The content of the text message
From	Input	This is the source/sender's phone number
SendStatusCollection	Return	Collection containing the send status for each recipient. See <a href="#">SendStatusCollection</a>

### ASP Example:

```
<%  
Set IntelliSMS = Server.CreateObject("IntelliSoftware.IntelliSMS")  
  
IntelliSMS.Username = "MyUsername"  
IntelliSMS.Password = "MyPassword"  
  
Set SendStatusColl = IntelliSMS.SendMessageToMultipleRecipients _  
("44771012345,44771023456", "Hello", "SENDER_ID")  
  
If SendStatusColl.OverallResult <> 1 Then  
    Summary = "Failed, Result=" & SendStatusColl.OverallResult  
Else  
    For Each Status In SendStatusColl  
        Summary = Summary & "To=" & Status.To & _  
            ", MessageId=" & Status.MessageId & _  
            ", Result=" & Status.ResultCode & "<BR>"  
    Next  
End If  
  
Response.Write Summary  
%>
```

## VB Example:

```
Dim IntelliSMS As New INTELLISMSLib.IntelliSMS

IntelliSMS.Username = "MyUsername"
IntelliSMS.Password = "MyPassword"

Dim SendStatusColl As INTELLISMSLib.SendStatusCollection

Set SendStatusColl = IntelliSMS.SendMessageToMultipleRecipients _
    ("44771012345,44771023456", "Hello", "SENDER_ID")

Dim Summary As String

If SendStatusColl.OverallResult <> ResultCodes.rcOK Then
    Summary = "Failed, Result=" & SendStatusColl.OverallResult
Else
    Dim Status As INTELLISMSLib.SendStatus
    For Each Status In SendStatusColl
        Summary = Summary & "To=" & Status.To & _
            ", MessageId=" & Status.MessageId & _
            ", Status=" & Status.ResultCode & vbCrLf
    Next
End If

MsgBox Summary
```

### 3.3. Send WAP Push Message

**SendWapPushMessage ( To As String, Title As String, HRef As String, From As String ) As SendStatusCollection**

#### Description:

Sends a WAP PUSH message via the Internet Gateway. WAP PUSH messages can send multimedia content, such as an Image, Wallpaper, Ringtone or a link to WAP website.

#### Parameters:

To	Input	This is the destination phone number(s) NOTE: Separate multiple numbers by a comma
Title	Input	Display Text for WAP PUSH message
HRef	Input	URL link to resource ( should start with 'http://' or 'https://' )
From	Input	This is the source/sender's phone number
SendStatusCollection	Return	Collection containing the send status for each recipient. See <a href="#">SendStatusCollection</a>

#### ASP Example:

```
<%  
Set IntelliSMS = Server.CreateObject("IntelliSoftware.IntelliSMS")  
  
IntelliSMS.Username = "MyUsername"  
IntelliSMS.Password = "MyPassword"  
  
Set SendStatusColl = IntelliSMS.SendWapPushMessage ( _  
    "44771012345,44771023456", "The message text", _  
    "http://www.intellisoftware.co.uk/images/IntelliSMS.gif", _  
    "SENDER_ID" )  
  
If SendStatusColl.OverallResult <> 1 Then  
    Summary = "Failed, Result=" & SendStatusColl.OverallResult  
Else  
    For Each Status In SendStatusColl  
        Summary = Summary & "To=" & Status.To & _  
            ", MessageId=" & Status.MessageId & _  
            ", Result=" & Status.ResultCode & "<BR>"  
    Next  
End If  
  
Response.Write Summary  
%>
```



## VB Example:

```
Dim IntelliSMS As New INTELLISMSLib.IntelliSMS

IntelliSMS.Username = "MyUsername"
IntelliSMS.Password = "MyPassword"

Dim SendStatusColl As INTELLISMSLib.SendStatusCollection

Set SendStatusColl = IntelliSMS.SendWapPushMessage ( _
    "44771012345,44771023456", "The message text", _
    "http://www.intellissoftware.co.uk/images/IntelliSMS.gif", _
    "SENDER_ID")

Dim Summary As String

If SendStatusColl.OverallResult <> ResultCodes.rcOK Then
    Summary = "Failed, Result=" & SendStatusColl.OverallResult
Else
    Dim Status As INTELLISMSLib.SendStatus
    For Each Status In SendStatusColl
        Summary = Summary & "To=" & Status.To & _
            ", MessageId=" & Status.MessageId & _
            ", Status=" & Status.ResultCode & vbCrLf
    Next
End If

MsgBox Summary
```

## Considerations:

Our gateway will only send one SMS message per submission. Therefore a WAP PUSH message is charged at 1 credit.

An INVALID\_REQUEST will result if the submitted text and href parameters would require 2 SMS messages to be sent. As a guide, the href and text parameters combined should not exceed 114 characters (assuming href starts with 'http://www.')

### 3.4. Send Binary Message

**SendBinaryMessage ( To As String, UserDataHeaderHex As String, UserDataHex As String, From As String ) As SendStatusCollection**

**Description:**

Sends a Binary SMS message via the Internet Gateway. Binary Messages can be used to send SmartMessages, such as Ringtones and Logos.

**Parameters:**

To	Input	This is the destination phone number(s) NOTE: Separate multiple numbers by a comma
UserDataHeaderHex	Input	(Optional) User Data Header
UserDataHex	Input	User Data (140 octets max)
From	Input	This is the source/sender's phone number
SendStatusCollection	Return	Collection containing the send status for each recipient. See <a href="#">SendStatusCollection</a>

**ASP Example:**

```
<%
Set IntelliSMS = Server.CreateObject("IntelliSoftware.IntelliSMS")

IntelliSMS.Username = "MyUsername"
IntelliSMS.Password = "MyPassword"

Set SendStatusColl = IntelliSMS.SendBinaryMessage ( _
    "44771012345,44771023456", "06050415820000", _
    "32F40100480E0100000000000000000000000000000000000000000000000000" & _
    "00000000000000000000000012478030000000E0080120CC06400000" & _
    "40080120CC06C00000479E7124F0EFFADCF64448892479B6DAC6" & _
    "CD4448F9241DB6DACECF44488124CDB6CD96CC44488924CDB6CD" & _
    "96CDE446712478E66D9EC600000000000000000000000000000000" & _
    "000000", "SENDER_ID" )

If SendStatusColl.OverallResult <> 1 Then
    Summary = "Failed, Result=" & SendStatusColl.OverallResult
Else
    For Each Status In SendStatusColl
        Summary = Summary & "To=" & Status.To & _
            ", MessageId=" & Status.MessageId & _
            ", Result=" & Status.ResultCode & "<BR>"
    Next
End If

Response.Write Summary
%>
```

## VB Example:

```
Dim IntelliSMS As New INTELLISMSLib.IntelliSMS

IntelliSMS.Username = "MyUsername"
IntelliSMS.Password = "MyPassword"

Dim SendStatusColl As INTELLISMSLib.SendStatusCollection

Set SendStatusColl = IntelliSMS.SendBinaryMessage ( _
    "44771012345,44771023456", "06050415820000", _
    "32F40100480E0100000000000000000000000000000000000000" & _
    "00000000000000000000000012478030000000E0080120CC06400000" & _
    "40080120CC06C00000479E7124F0EFFADCF64448892479B6DAC6" & _
    "CD4448F9241DB6DACECF44488124CDB6CD96CC44488924CDB6CD" & _
    "96CDE446712478E66D9EC60000000000000000000000000000000" & _
    "000000", "SENDER_ID" )

Dim Summary As String

If SendStatusColl.OverallResult <> ResultCodes.rcOK Then
    Summary = "Failed, Result=" & SendStatusColl.OverallResult
Else
    Dim Status As INTELLISMSLib.SendStatus
    For Each Status In SendStatusColl
        Summary = Summary & "To=" & Status.To & _
            ", MessageId=" & Status.MessageId & _
            ", Status=" & Status.ResultCode & vbCrLf
    Next
End If

MsgBox Summary
```

The above examples send an 'IntelliSoftware' operator logo to a handset.

### 3.5. Send Unicode Message

**SendUnicodeMessage ( To As String, Message As String, From As String ) As SendStatusCollection**

**SendUnicodeMessageHex ( To As String, MessageHex As String, From As String ) As SendStatusCollection**

#### Description:

Sends a Unicode SMS message to multiple recipients via the Internet Gateway. The Unicode format is used to send multilingual messages not support by the standard GSM character set.

See [Unicode Character Codes](#) for more details.

#### Parameters:

To	Input	This is the destination phone number(s) NOTE: Separate multiple numbers by a comma
Message	Input	The content of the text message (70 Unicode characters max.)
MessageHex	Input	Unicode text encoded in hexadecimal (140 octets max, 70 unicode characters)
From	Input	This is the source/sender's phone number
SendStatusCollection	Return	Collection containing the send status for each recipient. See <a href="#">SendStatusCollection</a>

#### ASP Example:

```
<%  
Set IntelliSMS = Server.CreateObject("IntelliSoftware.IntelliSMS")  
  
IntelliSMS.Username = "MyUsername"  
IntelliSMS.Password = "MyPassword"  
  
Set SendStatusColl = IntelliSMS.SendUnicodeMessage  
    ("44771012345,44771023456", "Hello", "SENDER_ID")  
  
If SendStatusColl.OverallResult <> 1 Then  
    Summary = "Failed, Result=" & SendStatusColl.OverallResult  
Else  
    For Each Status In SendStatusColl  
        Summary = Summary & "To=" & Status.To & _  
            ", MessageId=" & Status.MessageId & _  
            ", Result=" & Status.ResultCode & "<BR>"  
    Next  
End If  
  
Response.Write Summary  
>%
```

## VB Example:

```
Dim IntelliSMS As New INTELLISMSLib.IntelliSMS

IntelliSMS.Username = "MyUsername"
IntelliSMS.Password = "MyPassword"

Dim SendStatusColl As INTELLISMSLib.SendStatusCollection

Set SendStatusColl = IntelliSMS.SendUnicodeMessage _
    ("44771012345,44771023456", "Hello", "SENDER_ID")

Dim Summary As String

If SendStatusColl.OverallResult <> ResultCodes.rcOK Then
    Summary = "Failed, Result=" & SendStatusColl.OverallResult
Else
    Dim Status As INTELLISMSLib.SendStatus
    For Each Status In SendStatusColl
        Summary = Summary & "To=" & Status.To & _
            ", MessageId=" & Status.MessageId & _
            ", Status=" & Status.ResultCode & vbCrLf
    Next
End If

MsgBox Summary
```

## 3.6. Get Credit Balance

### GetBalance ( Balance As Variant ) As ResultCodes

#### Description:

Obtains the available credits on an account.

#### Parameters:

Balance	Output	Number of remaining credits
ResultCode	Return	See <b>ResultCodes</b>

#### ASP Example:

```
<%  
Dim IntelliSMS  
Set IntelliSMS = Server.CreateObject  
("IntelliSoftware.IntelliSMS")  
  
IntelliSMS.Username = "MyUsername"  
IntelliSMS.Password = "MyPassword"  
  
Dim Balance  
  
ResultCode = IntelliSMS.GetBalance ( Balance )  
%>
```

#### VB Example:

```
Dim IntelliSMS As New INTELLISMSLib.IntelliSMS  
  
IntelliSMS.Username = "MyUsername"  
IntelliSMS.Password = "MyPassword"  
  
Dim Balance As Variant  
Dim Result As INTELLISMSLib.ResultCodes  
  
Result = IntelliSMS.GetBalance ( Balance )
```

## 4. Collections

### 4.1. SendStatusCollection

Collection containing the send status for each recipient.

#### Methods:

Item(Index) As SendStatus	Returns an individual <b>SendStatus</b> object from the collection
---------------------------	--

#### Properties:

OverallResult	Indicates the overall success/failure. See <b>ResultCodes</b>
Count	Number of SendStatus objects in the collection

#### ASP Example:

```
<%
Set IntelliSMS = Server.CreateObject("IntelliSoftware.IntelliSMS")

IntelliSMS.Username = "MyUsername"
IntelliSMS.Password = "MyPassword"

Set SendStatusColl = IntelliSMS.SendMessageToMultipleRecipients _
("44771012345,44771023456", "Hello", "SENDER_ID")

If SendStatusColl.OverallResult <> 1 Then
    Summary = "Failed, Result=" & SendStatusColl.OverallResult
Else
    For Each Status In SendStatusColl
        Summary = Summary & "To=" & Status.To & _
            ", MessageId=" & Status.MessageId & _
            ", Result=" & Status.ResultCode & "<BR>"
    Next
End If

Response.Write Summary
%>
```

## VB Example:

```
Dim IntelliSMS As New INTELLISMSLib.IntelliSMS

IntelliSMS.Username = "MyUsername"
IntelliSMS.Password = "MyPassword"

Dim SendStatusColl As INTELLISMSLib.SendStatusCollection

Set SendStatusColl = IntelliSMS.SendMessageToMultipleRecipients _
    ("44771012345,44771023456", "Hello", "SENDER_ID")

Dim Summary As String

If SendStatusColl.OverallResult <> ResultCodes.rcOK Then
    Summary = "Failed, Result=" & SendStatusColl.OverallResult
Else
    Dim Status As INTELLISMSLib.SendStatus
    For Each Status In SendStatusColl
        Summary = Summary & "To=" & Status.To & _
            ", MessageId=" & Status.MessageId & _
            ", Status=" & Status.ResultCode & vbCrLf
    Next
End If

MsgBox Summary
```



## 5. Enumerations

### 5.1. ResultCodes Enumeration

#### INTELLISMLib.ResultCodes

##### Description:

Indicates the success/failure of a request.

Returned by SendMessage, GetBalance, SendStatus.ResultCode and SendStatusCollection.OverallResult properties.

##### Values:

0	rcUnknown	Unknown error has occurred
1	rcOK	Request was successfully actioned
2	rcNoUsername	Username was not supplied (Set Username property)
3	rcNoPassword	Password was not supplied (Set Password property)
4	rcNoTo	Message recipient(s) not specified
5	rcNoText	Message text not specified
6	rcLoginInvalid	Username/Password not valid
7	rcInsufficientCredit	Insufficient credit available to complete the request
8	rcGatewayError	An error occurred on remote gateway (please retry later)
9	rcInternalError	An error occurred on remote gateway (please retry later)
10	rcHTTPConnectionError	Unable to make connection to remote gateway
11	rcInvalidNumber	The recipient number is invalid
12	rcInvalidRequest	The request was invalid
13	rcServerTooBusy	Remote gateway too busy (please retry later)
14	rcMsgIdInvalid	Supplied Message ID not valid
15	rcParameterMissing	Mandatory parameter is missing
16	rcParameterInvalid	A supplied parameter is invalid
17	rcAccountExists	CreateSubaccount - subaccount already exists

### 5.2. SendStatus Enumeration

Provides the send status for a recipient.

##### Properties:

To	Recipient's phone number that this status report is relevant
MessageId	ID for this message.
ResultCode	Indicates the success/failure of message submission for this recipient. See <b>ResultCodes</b>

## 6. Advanced Features

### 6.1. Send to Distribution Lists (Group Send)

Our gateway allows you to send SMS messages to preconfigured Distribution Lists or Groups. To send to a Distribution List or Group simply insert the Group Name where you would normally place the recipient's phone number. Distribution Lists or Groups are setup in your online account login on the Contacts page.

### 6.2. Reply Tracking

The IntelliSoftware platform provides tracking of SMS replies, allowing you to determine which particular message a mobile user is replying to. When you submit a message to the SMS Gateway you can supply a User Context parameter. When a reply is received, the IntelliSoftware platform will include the User Context parameter when forwarding the message to your server.

The User Context can be supplied to the IntelliSoftware platform in the following ways:

.Net Component	use SendMsgWithUserContext method
COM Component	use SendMessageWithUserContext method
HTTP Interface	add 'usercontext' POST/GET parameter
PHP SDK	use SendMessageWithUserContext method
Java SDK	use SendMessageWithUserContext method

The IntelliSoftware platform will forward the User Context for received messages in the following ways:

HTTP Interface	'usercontext' POST/GET parameter
----------------	----------------------------------