



**IntelliSoftware**  
**SMS Gateway - DotNet Component Interface**

Version 1.4

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. DotNet Component Interface Overview
3. Submitting to our gateway
  - 3.1. Send Text Message
  - 3.2. Send Text Message to Multiple Recipients
  - 3.3. Send MMS Message
  - 3.4. Send WAP Push Message
  - 3.5. Send Binary Message
  - 3.6. Send Unicode Message
  - 3.7. Get Credit Balance
4. Collections
  - 4.1. SendStatusCollection
5. Classes
  - 5.1. MMSMessage
  - 5.2. MessagePart
6. Exception
  - 6.1. IntelliSMSEException
7. Advanced Features
  - 7.1. Send to Distribution Lists (Group Send)
  - 7.2. 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	19/10/2006	Chris Vine	Added Unicode Support and Advanced Features
1.3	18/02/2009	Chris Vine	Added MMS Messaging
1.4	27/04/2009	Chris Vine	Updated Company Address

## 2. DotNet Component Interface Overview

You can use our dotnet component to send message from an ASP.NET web page or Windows Form Application. This component is free to download and use. You can distribute this component with your own SMS enabled applications.

### Methods:

<b>SendMsg</b>	Sends a standard SMS text message.
<b>SendMsgToMultipleRecipients</b>	Sends a standard SMS text message to multiple recipients.
<b>SendMMSMessage</b>	Sends an MMS Message to one or more recipients.
<b>SendWapPushMsg</b>	Sends a WAP PUSH message. WAP PUSH messages can be used to send multimedia content to mobile phones.
<b>SendBinaryMsg</b>	Sends a Binary Message. Binary Messages can be used to send SmartMessages, such as Ringtones and Logos.
<b>SendUnicodeMsg</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

## 3. Submitting to our gateway

### 3.1. Send Text Message

**String SendMsg ( String To, String Text, String From )**

**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	Return	Unique message ID

**Exceptions:**

<b>IntelliSMSException</b>	Error returned by the IntelliSoftware SMS gateway
----------------------------	---

**C# Example:**

```
using IntelliSoftware;

IntelliSMS objIntelliSMS = new IntelliSMS();

objIntelliSMS.Username = "MyUsername";
objIntelliSMS.Password = "MyPassword";

String MessageId = objIntelliSMS.SendMsg
    ( "44771012345", "Hello", "SENDER_ID" );
```

## 3.2. Send Text Message to Multiple Recipients

**SendStatusCollection SendMsgToMultipleRecipients ( StringCollection To, String Text, String From )**

### Description:

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

### Parameters:

To	Input	List of destination phone numbers
Text	Input	The content of the text message
From	Input	This is the source/sender's phone number
<b>SendStatusCollection</b>	Return	This collection is populated with the send status for each recipient

### Exceptions:

<b>IntelliSMSException</b>	Error returned by the IntelliSoftware SMS gateway
----------------------------	---

### C# Example:

```
using IntelliSoftware;

IntelliSMS objIntelliSMS = new IntelliSMS();

objIntelliSMS.Username = "MyUsername";
objIntelliSMS.Password = "MyPassword";

StringCollection ToList = new StringCollection();
ToList.Add ( "44771012345" );
ToList.Add ( "44771054321" );

SendStatusCollection SendStatusList;

SendStatusList = objIntelliSMS.SendMsgToMultipleRecipients
    ( ToList, "Hello", "SENDER_ID" );
```

### 3.3. Send MMS Message

**SendStatusCollection SendMMSMessage ( StringCollection To, MMSMessage Message )**

#### Description:

Sends an MMS message to one or more recipients via the Internet Gateway.

#### Parameters:

<b>To</b>	Input	List of destination phone numbers
<b>MMSMessage</b>	Input	MMS Message to be sent (includes subject and message parts).
<b>SendStatusCollection</b>	Return	This collection is populated with the send status for each recipient.

#### Exceptions:

<b>IntelliSMSException</b>	Error returned by the IntelliSoftware SMS gateway
<b>IntelliSMSClientSideException</b>	Error raised by the IntelliSMS Component (Client-Side) prior to making the submission to the IntelliSoftware SMS gateway

#### C# Example:

```
using IntelliSoftware;

MMSMessage NewMMSMessage = new MMSMessage();
NewMMSMessage.Subject = "MMS Message Subject";

MessagePart MessagePart1 = new MessagePart();
MessagePart1.Filename = "Image1.gif";
MessagePart1.SourceFilename = "C:\Image1.gif";
NewMMSMessage.MessageParts.Add ( MessagePart1 );

MessagePart MessagePart2 = new MessagePart();
MessagePart2.Filename = "Text1.txt";
MessagePart2.Text = "This is a text part";
NewMMSMessage.MessageParts.Add ( MessagePart2 );

IntelliSMS objIntelliSMS = new IntelliSMS();

objIntelliSMS.Username = "MyUsername";
objIntelliSMS.Password = "MyPassword";

StringCollection ToList = new StringCollection();
ToList.Add ( "44771012345" );
ToList.Add ( "44771054321" );

SendStatusCollection SendStatusList;

SendStatusList = objIntelliSMS.SendMMSMessage
( ToList, NewMMSMessage );
```

### 3.4. Send WAP Push Message

**SendStatusCollection SendWapPushMsg ( StringCollection To, String Title, String HRef, String From )**

#### 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	List of destination phone numbers
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
<b>SendStatusCollection</b>	Return	This collection is populated with the send status for each recipient

#### Exceptions:

<b>IntelliSMSEException</b>	Error returned by the IntelliSoftware SMS gateway
-----------------------------	---

#### C# Example:

```
using IntelliSoftware;

IntelliSMS objIntelliSMS = new IntelliSMS();

objIntelliSMS.Username = "MyUsername";
objIntelliSMS.Password = "MyPassword";

StringCollection ToList = new StringCollection();
ToList.Add ( "44771012345" );

SendStatusCollection SendStatusList;

SendStatusList = objIntelliSMS.SendWapPushMsg (
    ToList, "The message text",
    "http://www.intellisoftware.co.uk/images/IntelliSMS.gif",
    "SENDER_ID" );
```

#### Considerations:

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

An InvalidRequest 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.') )





### 3.6. Send Unicode Message

**SendStatusCollection SendUnicodeMsg ( StringCollection To, String Message, String From )**

**SendStatusCollection SendUnicodeMsgHex ( StringCollection To, String MessageHex, String From )**

#### 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	List of destination phone numbers
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
<b>SendStatusCollection</b>	Return	This collection is populated with the send status for each recipient

#### Exceptions:

<b>IntelliSMSException</b>	Error returned by the IntelliSoftware SMS gateway
----------------------------	---

#### C# Example:

```
using IntelliSoftware;

IntelliSMS objIntelliSMS = new IntelliSMS();

objIntelliSMS.Username = "MyUsername";
objIntelliSMS.Password = "MyPassword";

StringCollection ToList = new StringCollection();
ToList.Add ( "44771012345" );
ToList.Add ( "44771054321" );

SendStatusCollection SendStatusList;

SendStatusList = objIntelliSMS.SendUnicodeMsg
    ( ToList, "Hello", "SENDER_ID" );
```

### 3.7. Get Credit Balance

#### Integer GetBalance ()

#### Description:

Obtains the available credits on an account.

#### Parameters:

Balance	Return	Number of remaining credits
---------	--------	-----------------------------

#### Exceptions:

<b>IntelliSMSException</b>	Error returned by the IntelliSoftware SMS gateway
----------------------------	---

#### C# Example:

```
using IntelliSoftware;  
  
IntelliSMS objIntelliSMS = new IntelliSMS();  
  
objIntelliSMS.Username = "MyUsername";  
objIntelliSMS.Password = "MyPassword";  
  
int Balance = objIntelliSMS.GetBalance();
```

## 4. Collections

### 4.1. SendStatusCollection

#### Description:

Contains the send status for each recipients.

#### C# Example:

```
using IntelliSoftware;

IntelliSMS objIntelliSMS = new IntelliSMS();

objIntelliSMS.Username = "MyUsername";
objIntelliSMS.Password = "MyPassword";

StringCollection ToList = new StringCollection();
ToList.Add ( "44771012345" );

SendStatusCollection SendStatusList;

SendStatusList = objIntelliSMS.SendMsgToMultipleRecipients
    ( ToList, "Hello", "SENDER_ID" );

string Report = "";
foreach ( SendStatus sendStatus in SendStatusList )
{
    Report += "To:" + sendStatus.To + ", " +
        "MessageId:" + sendStatus.MessageId + ", " +
        "ResultCode:" + sendStatus.ResultCode + "\n";
}

MessageBox.Show ( Report );
```

## 5. Classes

### 5.1. MMSMessage

class **MMSMessage**

**Description:**

Represents an MMS Message. Includes message subject and message parts

**Properties:**

<b>Subject</b>	String	MMS Message Subject
<b>MessageParts</b>	Collection	Collection containing message parts. Contains <b>MessagePart</b> objects.
<b>From</b>	String	This is the source/sender's phone number. Leave blank to receive replies. Can be numerical (e.g. long number or shortcode), alphanumeric sender's Id's not supported.

### 5.2. MessagePart

class **MessagePart**

**Description:**

Represents a Message Part in an MMS Message

**Properties:**

<b>Filename</b>	String	Filename of the message part (e.g. "Image.gif"). This is the Filename of the message part as it will appear in the MMS Message received on the phone. Do not include a file path.
<b>ContentType</b>	String	(Optional) Content type of the MMS Message Part (e.g. "image/gif"). If not supplied the Content Type will be determined based on the file extension given in Filename property.
<b>Text</b>	String	(Optional) Text content of a Text Message Part
<b>SourceFilename</b>	String	(Optional) Full path and filename of the file that contains the content of the MMS Message Part (e.g. "C:\Image.gif")
<b>SourceStream</b>	Stream	(Optional) An open stream that contains the content for the Message Part. NOTE: Caller is responsible for closing the stream after the call to SendMMSMessage.

**NOTE:** You must supply message content through the Text, SourceFilename or SourceStream property, if you do not set one of these properties an error will be returned.

## 6. Exception

### 6.1. IntelliSMSException

#### Description:

Represents error returned by the IntelliSoftware SMS Internet gateway.

#### Properties:

ResultCode	Result code: 1 - OK 2 - NoUsername 3 - NoPassword 4 - NoTo 5 - NoText 6 - LoginInvalid 7 - InsufficientCredit 8 - GatewayError 9 - InternalError 10 - HTTPConnectionError 11 - InvalidNumber 12 - InvalidRequest 13 - ServerTooBusy 14 - MsgIdInvalid
InnerException	Provide additional information for a HTTPConnectionError

## 7. Advanced Features

### 7.1. 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.

### 7.2. 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
----------------	----------------------------------