Now SMS/MMS Gateway
2009 Edition
http://www.nowsms.com

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved
Publication Date: June 30, 2009

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved
Publication Date: June 30, 2009

NOW SMS/MMS GATEWAY....1

SMS and MMS Protocols Implemented by NowSMS....3
MMS Gateway Connectivity (client protocols)....3
SMS Gateway Connectivity (client protocols)....3
Application Connectivity for MMS Submission (server protocols)....3
Application Connectivity for SMS Submission (server protocols)....3
Examples of How Customers Use NowSMS....4

NOWSMS QUICK START: BASIC CONFIGURATION....6

Need Help?....7

INSTALLING NOWSMS....8

System Requirements....8
Software Installation....9
New Customer: Installing Free 60-Day Trial Version....11
New Customer: Applying a Purchased License to an existing Trial Installation....12
New Customer: Installing a Purchased License Version....15
Existing Customer: Upgrading a NowSMS Installation....18
Existing Customer: Moving NowSMS to a new Computer....21

CONFIGURING SMSC CONNECTIONS....24

GSM Modems....25
GSM Modem Troubleshooting Tips....29
SMPP SMSC....34
HTTP SMSC....40
Receiving SMS Messages via an HTTP SMSC Connection....44
UCP/EMI SMSC....45
CIMD2 SMSC....48
Additional SMSC Configuration Options....50
Routing Options....52
SMS Message Routing Logic....54

RUNNING AS A SERVICE....55

Event Log and System Alerts....58

CONFIGURING THE WEB INTERFACE AND SMPP SERVER....60

SMPP Server Options....63
Defining SMS User Accounts....64

NOWSMS IN HIGH AVAILABILITY/LOAD BALANCED ENVIRONMENTS....70

WEB MENU INTERFACE....73

Send Text Message....75
Send EMS Message....77
Send EMS Text Message....78
Send EMS ring tone....80
Send EMS Picture Message....83
Send Binary Message....84
Send Nokia Ring Tone....85
Send Nokia CLI (Group) Icon....86
Send Nokia Operator Logo....87
Send Nokia Picture Message....88

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send Binary Message Other....89
SEND WAP PUSH MESSAGE....90
Send WAP Push Advanced....92
SEND MMS MESSAGE....94
Digital Rights Management Options....95
SEND MMS NOTIFICATION....97
SEND MULTIMEDIA CONTENT MESSAGE....98
SEND WAP OTA SETTINGS....101
WAP OTA: Browser/MMS Client - GPRS/EDGE/Packet Data Settings....102
WAP OTA - Browser/MMS Client: GSM/Circuit Switch Data Settings....104
WAP OTA: Bookmark....106
WAP OTA: SyncML Settings....107
WAP OTA: Wireless Village/ IMPS Settings....110
SEND OMA OTA SETTINGS....112
Access Point Settings Section....114
Proxy Settings Section....114
Browser Settings Section....115
MMS Settings Section....115
Wireless Village/IMPS Section....115
SyncML DS Settings Section....116
E-Mail Settings Section....116
PIN/Security Section....117
SEND XML SETTINGS DOCUMENT....118
Nokia/Ericsson Over The Air Settings (OTA) Specification up to and including v7.1....118
OMA (Open Mobile Alliance) Provisioning Content....119
OMA (Open Mobile Alliance) DRM Rights Objects....119
WAP Push Service Indication, Service Load and Cache Operation....119
OMA (Open Mobile Alliance) E-Mail Notification (EMN)....119
OTA PIN and OTA PIN Type....119
SEND VOICE MAIL NOTIFICATION....121

MMSC MESSAGING SERVER....122

HOW MMS WORKS....122
CONFIGURING THE MMSC....126
E-MAIL – MMS GATEWAY....131
CONFIGURING MMS VASP ACCOUNTS....132
CONNECTING TO AN OPERATOR MMSC....136
Connecting to an Operator MMSC – Using a GPRS Modem....136
Connecting to an Operator MMSC – Sending MMS Messages....142
Sending MMS Message – MM1 (GPRS Modem)....145
Sending MMS Message – MM7....148
Sending MMS Message – MM4....151
Sending MMS Message – EAIF....153
Sending MMS Message – Direct Delivery....155
Sending MMS Message – Convert to WAP Push with Web Link....156
Sending MMS Message – Convert to SMS with Web Link....157
Sending MMS Message – Convert to SMS with Web Link (Direct)....158
Sending MMS Message – Convert to SMS with Web Link (Code)....159
Sending MMS Message – Block/Reject Message....162
Connecting to an Operator MMSC – Receiving MMS Messages....163

SUBMITTING MMS MESSAGES TO NOWSMS....165

SEND MMS MESSAGE WITH PHP....166
SEND MMS MESSAGE WITH JAVA....172

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send MMS Message from Command Line....176
Now SMS/MMS Proprietary URL Submission....178
MM7....180
MM4....183
MM1....185
EAIF....186

SUBMITTING SMS MESSAGES - URL PARAMETERS....187

Sending Text Messages....201
Sending EMS Messages....205
Sending EMS Messages – EMS Text....206
Sending EMS Messages – EMS Ring Tone....208
Sending EMS Messages – EMS Picture Message....211
Sending Binary Messages....213
Sending WAP Push Messages....215
Sending Multimedia WAP Push Messages....218
Sending WAP OTA Messages....220
WAP Bookmark OTA Messages....220
WAP Configuration OTA Messages....220
Sending OMA Provisioning Content OTA Messages....224
Sending XML Settings Documents and Objects....227
Nokia/Ericsson Over The Air Settings (OTA) Specification up to and including v7.1....227
OMA (Open Mobile Alliance) Provisioning Content....227
OMA (Open Mobile Alliance) DRM Rights Objects....227
WAP Push Service Indication, Service Load and Cache Operation....227
OMA (Open Mobile Alliance) E-Mail Notification (EMN)....228
HTTP POST Format....228
OTA PIN and OTA PIN Type....229
Sending MMS Notifications and Content....230
Creating MMS Message Files – MMSCOMP....231
Sending Voice Mail Notification Messages....234

2-WAY SMS SUPPORT....236

2-WAY MMS SUPPORT....240

E-MAIL TO SMS/MMS CONNECTIVITY....243

Configuring the SMTP Server....244
E-Mail Access Restrictions....246
Additional E-Mail Settings....248
Using the POP3 Server....249
Using POP3/SMTP with Outlook Express....250
Using POP3/SMTP with Mozilla Thunderbird....260
Receiving SMS or MMS Messages via POP3....271
Routing SMS Messages to a POP3 Client....271
Routing MMS Messages to a POP3 Client....274
Routing SMS Messages to E-Mail....276
Routing MMS Messages to E-Mail....277

SMS GATEWAY BILLING AND CHARGING: ACCOUNTING CALLBACKS....279

SMSSend PreAuth Callback....280
SMSSend Accounting Callback....281

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSC BILLING AND CHARGING: ACCOUNTING CALLBACKS....282

MMSSend PreAuth Callback....283
MMSSend Charging Callback....285
MMSRetrieve Accounting Callback....287
MMSOut Accounting Callback....288
MMSOutFailed Accounting Callback....289
MMSDeliveryReport PreAuth Callback....290
MMSDeliveryReport Charging Callback....291
MMSReadReport PreAuth Callback....292
MMSReadReport Charging Callback....293
MMSEMail PreAuth Callback....294
MMSEMail Charging Callback....295

QUERY NOWSMS SERVER STATUS....296

INTERFACING WITH NOWSMS VIA PHP....299

Send SMS Text Message with PHP....300
Send OMA Client Provisioning with PHP....302
Send MMS Message with PHP....305
Receive MMS Message with PHP....305

INTERFACING WITH NOWSMS VIA JAVA....306

Send SMS Text Message with Java....307
Send MMS Message with Java....312

INTERFACING WITH NOWSMS VIA COMMAND LINE INTERFACE....313

Send SMS Text Message from the Command Line....313
Send MMS Message from the Command Line....313
Send WAP Push and Binary SMS from the Command Line....314
Send OMA Client Provisioning from the Command Line....317

DIGITAL RIGHTS MANAGEMENT....320

DRM with the NowSMS Web Interface....321
DRM when interfacing with NowSMS Programmatically....322
DRM Forward Lock and Combined Delivery....322
DRMCOMP Utility & DRM Separate Delivery....325
DRMCOMP – Forward Lock....325
DRMCOMP – Combined Delivery....326
DRMCOMP – Separate Delivery....329

ADVANCED CONFIGURATION SETTINGS....332

SMSGW.INI Parameters....333
SMSGW.INI, [SMSGW] section: 333
SMSGW.INI, section specific to an SMSC definition: 338
SMSGW.INI, [SMPPOptions] section: 342
SMSGW.INI, [SMPP] section: 344
SMSGW.INI, [UCP] section: 344
MMSC.INI Parameters....345
MMSC.INI, [MMSC] section: 345
MMSC.INI, [ShortCode] section: 356
MMSC.INI, [IPNotify] section: 356
MMSC.INI, [HostNameSenderOverride] section: 356

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSC.INI, [TranslateText] section: *357*
MMSC Outbound Routing Parameters 357
VASP.INI, [VASP] section: *357*

TECHNICAL BULLETINS 359

Now SMS/MMS Accounting Callbacks 360
MMS Accounting Callbacks 360
MMSSend PreAuth Callback 360
MMSSend Accounting Callback 360
MMSEMail Pre-Auth Callback 361
MMSEMail Accounting Callback 361
MMSRetrieve Accounting Callback 361
SMS Accounting Callbacks 361
SMSSend PreAuth Callback 362
SMSSend Accounting Callback 362
Now MMSC Operator Considerations 363
Configuring the NowWAP Proxy to Forward MSISDN 364
2-Way SMS Returning a Non-Text Response 368
Using Port 80 on a PC running IIS 370
Provisioning MMSC User Accounts via HTTP 371
Faster GSM Modem Speeds with SMS over GPRS 372
Routing MMS Notifications via a WAP Push Proxy Gateway 374
Sending SMS from the Command Line 376
WAP Push or OTA: Unknown Sender 378
Using NowSMS as an MMSC in CDMA or CDMA2000 Environments 379
Delivering MMS Notifications with WAP Push over IP 382
NowSMS as a WAP Push Proxy Gateway 383
Provisioning SMS and MMSC User Accounts via HTTP 385
MMS Virus Blocking 387
NowSMS v5.51b – March 2005 387
NowSMS 2006 – March 2006 388
NowSMS and OMA Digital Rights Management (DRM) 390
Receiving MMS Messages with a PHP Script: HTTP File Upload Post 391
Receiving MMS via HTTP File Upload Post 391
PHP Processing of MMS HTTP File Upload Post (mmsreceive.php) 393
Notes about installing/configuring PHP and testing the PHP script 396
Configuring NowSMS to call your PHP Script 397

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Now SMS/MMS Gateway

The Now SMS/MMS Gateway (NowSMS) is an SMS and MMS content delivery solution. NowSMS is a fast track to deploying and developing SMS, MMS and WAP Push and OTA solutions.

NowSMS is an easy-to-install SMS Gateway, MMS Gateway, WAP Push Proxy Gateway and Multimedia Messaging Center (MMSC). NowSMS is a scalable solution that is affordable for development, testing, with scalability to support full production mobile operator systems.

For additional technical information about the Now SMS/MMS Gateway, please visit our web site at http://www.nowsms.com.

NowSMS includes the following features:

- Supports SMS connectivity via one or more GSM modems (or GSM phones connected to a PC serial port), or over TCP/IP connections using SMPP, UCP/EMI and/or HTTP protocols.

- Supports least cost routing with pattern matching to route messages to different SMS connections based on destination.

- Supports sending and receiving MMS messages either via direct SMS/WAP delivery with its built-in MMSC, or can interface to operator MMSCs using the MM1, MM4 (SMTP), MM7 (XML-SOAP based HTTP POST API) and/or EAIF (Nokia proprietary API) protocols. The MM1 interface can talk to an operator MMSC over a GPRS/GSM modem without a special operator account.

- Supports easy generation and delivery of MMS messages, and includes an MMS compiler for generating the binary headers and message formats required for MMS content.

- Includes a powerful MMSC for processing MMS messages, supporting MMS versions 1.0 thru 1.3. The MMSC supports dynamic content adaptation and content conversion to help simplify the process of delivering MMS content to devices with varied characteristics. The MMSC also includes a built-in SMTP e-mail gateway for bi-directional exchange of messages between MMS compatible devices and internet e-mail recipients.

- Supports the MM1, MM4, MM7 and EAIF protocols to allow applications and Value Added Service Providers (VASPs) to send and receive MMS messages via the gateway.

- Supports Unicode (UTF-8) formats for both SMS and MMS messages, enabling deployment in multilingual environments.

- Supports easy generation and delivery of WAP Push messages, independent of the WAP gateway being used.

- Supports Multimedia WAP Push to simplify the delivery of multimedia objects and Java applications via WAP Push.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

- Supports ("Over The Air") configuration settings and bookmarks, including support for the Open Mobile Alliance (OMA) Provisioning Content v1.1.

- Supports 2-way SMS for interactive application development. SMS messages received by the gateway can trigger either an executable program to be run, or an HTTP request. Simple text responses back to the user can be returned as output of the request. More complex responses, including MMS or other binary SMS content, are also supported.

- Supports 2-way MMS for interactive multimedia application development. MMS messages received by the gateway are parsed into individual file components that can be easily processed by a user supplied tools. For example, received MMS images could be automatically posted to a web site.

- Includes an SMPP server, simplifying the process of connecting multiple gateways and applications.

- Provides an SMTP interface with SMTP Authentication support, allowing a user account to login via SMTP with an e-mail client to submit bulk delivery of SMS or MMS messages.

- Supports sending of other binary SMS formats, including EMS, ring tones, etc.

- Supports concatenated SMS for SMS text messages longer than 160 characters.

- Supports easy generation and delivery of new voice mail notification messages, simplifying the integration of office voice mail with mobile voice mail.

- Supports Open Mobile Alliance Digital Rights Management (OMA DRM) v1.0 with support for forward-lock, combined delivery and separate delivery DRM message types.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

SMS and MMS Protocols Implemented by NowSMS

MMS Gateway Connectivity (client protocols)

- ✔ MM7
- ✔ Ericsson's proprietary MM7 Variation
- ✔ LogicaCMG proprietary PAP/MM7
- ✔ Materna AnnyWay's proprietary MM7 Variation
- ✔ MM4
- ✔ MM3 (SMTP)
- ✔ MM1
- ✔ EAIF (Nokia proprietary MM1)
- ✔ MM1 over a GPRS modem

SMS Gateway Connectivity (client protocols)

- ✔ SMPP
- ✔ UCP/EMI
- ✔ CIMD2
- ✔ HTTP
- ✔ GSM Modem (ETSI GSM 07.05 / 3GPP TS 23.005)

Application Connectivity for MMS Submission (server protocols)

- ✔ MM7
- ✔ Ericsson's proprietary MM7 Variation
- ✔ LogicaCMG proprietary PAP/MM7
- ✔ Materna AnnyWay's proprietary MM7 Variation
- ✔ MM4
- ✔ MM3 (SMTP)
- ✔ MM1
- ✔ EAIF (Nokia proprietary MM1)

Application Connectivity for SMS Submission (server protocols)

- ✔ SMPP
- ✔ HTTP
- ✔ SMTP

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Examples of How Customers Use NowSMS

NowSMS is an extremely powerful and flexible tool. It is unusual for any one NowSMS installation to make use of all of the features of NowSMS. It is more typical that customers will use only one, or a few, of the many features of NowSMS.

Below is a list of examples of how different customers use NowSMS:

- **SMS Gateway** – NowSMS can manage simultaneous connections to one or more SMSCs, supporting the major SMSC protocols, including SMPP, UCP/EMI, CIMD2, HTTP, and/or GSM modems. NowSMS handles the low level protocol details, and makes it easy to switch between different providers, as well as making it easy to add additional connections for situations where it is advantageous to route messages for different countries via different providers.

- **2-Way SMS and MMS Application Enabler** – NowSMS is a great tool for enabling rapid development of interactive SMS applications and services. When NowSMS receives an SMS message, it can be configured to dispatch that message to a script running on an HTTP server, to a local executable program, or local script or batch file. This provides a simple way to get received messages into an application, so that the application can perform custom processing on the message. The application can generate a simple reply back to the received message, or perform more advanced application specific logic.

- **MMSC** – NowSMS is an Multimedia Messaging Service Centre, or MMSC. It has the capability to support person-to-person (phone-to-phone) MMS messaging, as well as application-to-person and person-to-application MMS messaging. Over 30 mobile operators are using NowSMS as an MMSC to enable multimedia messaging on their networks.

- **MMSC for Application to person messaging** – In many environments, NowSMS is implemented as a secondary MMSC to enable application to person MMS messaging, while another MMSC provides person-to-person MMS messaging services.

- **MMS Gateway** – NowSMS can interface to existing MMSCs using MM7, MM4, or EAIF. It can also interface to operator MMSCs over a GPRS modem using MM1. NowSMS handles the low level protocol details, and makes it easy to switch between different providers, as well as making it easy to add additional connections. NowSMS even supports many of the non-standard protocol variations deployed by major MMSC vendors.

- **WAP Push Proxy Gateway (PPG)** – NowSMS makes it easy to send WAP Push messages, supporting push message submission via a simple HTTP interface, as well as support for the more advanced Push Access Protocol (PAP) interface. NowSMS also includes support for Multimedia WAP Push, which can be used as a lower cost alternative to MMS for the delivery of multimedia content.

- **OTA Provisioning Gateway** – NowSMS supports OTA (Over-the-Air) provisioning based upon the Nokia/Ericsson OTA specification, and the Open Mobile Alliance (OMA) Provisioning Content specification. These protocols allow browser, messaging and synchronization settings to be sent to devices over SMS. OMA

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Provisioning also supports additional setting types, such as those used for Push-to-Talk, e-mail, and more.

- **Digital Rights Management (DRM) Enabler** – NowSMS provides a simple interface for sending out MMS or WAP Push messages with DRM Forward-Lock enabled. NowSMS also supports the encoding of more advanced DRM rights messages by supporting OMA DRM 1.0 with full support for both combined delivery and separate delivery. NowSMS also supports ROAP Trigger messages (root XML element <roap:roapTrigger>) as defined in the OMA DRM 2.1 specification.

- **SMSC Multiplexer** – NowSMS allows multiple applications to share SMSC connections by functioning as an SMPP server. Applications designed to submit messages using SMPP can connect to a NowSMS server, and NowSMS can route the messages over any of its supported SMSC links. NowSMS can also support multiple clients connecting in to submit messages over an HTTP/Web interface.

- **E-Mail to SMS or MMS Gateway** - NowSMS can be configured to provide bi-directional e-mail to SMS and MMS support.

- **Testing Tool** - Because NowSMS supports a wide range of protocols, NowSMS is a popular tool for device manufacturers and testing labs.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

NowSMS Quick Start: Basic Configuration

NowSMS is an extremely powerful and flexible tool. However, with power and flexibility comes complexity.

The purpose of this section is to help get you started with NowSMS. Different installations will want to use different features and functionality of NowSMS. However, before exploring some of the more advanced functionality, it is a good idea to configure basic functionality.

This is particularly important, because many people do not understand how SMS or MMS works.

NowSMS is not a bulk SMS service provider, and it is not a replacement for an SMS service provider.

In order to send any SMS or MMS messages with NowSMS, you must either have an account with an SMS service provider, or a GSM modem.

The Now SMS/MMS Gateway is a middleware tool for SMS, not a replacement for an SMS service provider.

The basic steps for configuring NowSMS are detailed below.

1.) Install the NowSMS Software – The installation of the NowSMS software is fairly straight-forward, however if you need more information on this process, please see **Installing NowSMS** on page 8.

2.) **Configure the SMSC Connection** - The Now SMS/MMS Gateway requires a connection to an SMSC (Short Messaging Service Centre) to interface with SMS and MMS networks. An SMSC connection can be a GSM modem, or it can be a connection to an SMS service provider over the internet or a private network using one of the following protocols: SMPP, UCP/EMI, CIMD2 or HTTP. For information on configuring an SMSC connection, please refer to **Configuring SMSC Connections** beginning on page 24.

3.) **Configure the Web Interface Settings of NowSMS** - When you wish to submit any type of SMS message, including MMS or WAP push messages, you must submit the request via the NowSMS web interface. The "Web" configuration dialog contains settings relevant to this web interface. For information on configuring the web interface, please refer to **Configuring the Web Interface** beginning on page 60.

4.) **Configure NowSMS to run as a service** - The next step in configuring the SMS/MMS Gateway is to install the service. The Now SMS/MMS Gateway installs as a pair of Windows services, which means that it is automatically loaded and run when the computer is restarted. The service can also be manually started and stopped via the Services option in the Windows control panel. To install the Now SMS/MMS Gateway as a service, use the "Service" configuration dialog. For more information on this step, please refer to **Running as a Service** beginning on page 55.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

5.) **Send an SMS Message** – Verify that your SMSC connection is functioning properly. Try sending an SMS message via the **Web Menu Interface** *(see page 73)*.

6.) **Optional: Configure MMSC Connectivity** – If you plan on using any of the MMS related functionality of NowSMS, read **How MMS Works** on page 122. This section explains how MMS works, and the two primary ways in which NowSMS can be configured to send MMS messages, either as a direct delivery MMSC or via a connection to an operator MMSC.

7.) **Optional: Send an MMS Message** – Verify that your MMSC connectivity is functioning properly. Try sending an MMS message via the **Web Menu Interface** *(see page 94)*.

Need Help?

If you can't find an answer to your question in this manual, please visit our web site at http://www.nowsms.com. In particular, the Discussion Board area of our web site includes answers and discussions from others, and can be an excellent resource for more information.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Installing NowSMS

System Requirements

To install the Now SMS/MMS Gateway software, you will need a PC running Windows XP, Windows Vista, Windows 2003 Server or Windows 2008 Server. Server, workstation, business and home editions of these operating systems are supported.

The Now SMS/MMS Gateway also requires a connection to an SMSC (Short Messaging Service Centre), or SMS service provider, to interface with SMS and MMS networks. An SMSC connection can consist of one or more of the following:

GSM Modem – A GSM modem or phone connected to a PC serial port (or to a USB port with an appropriate modem driver).

SMPP (Short Message Peer to Peer Protocol) – A TCP/IP connection over the internet or a private network to a service that supports v3.3 or v3.4 of the SMPP protocol.

UCP/EMI (Universal Computer Protocol / External Machine Interface) – A TCP/IP connection over the internet or a private network to a service that supports v3.5 or higher of the UCP/EMI protocol. UCP/EMI is primarily implemented by CMG SMSCs.

CIMD2 (Computer Interface to Message Distribution, version 2) – A TCP/IP connection over the internet to a service that supports the CIMD2 protocol. CIMD2 is implemented by Nokia SMSCs.

HTTP (Hyper Text Transport Protocol, e.g., the standard protocol for the "web") – A TCP/IP connection over the internet or private network to a service that accepts SMS messages via an HTTP "GET" based protocol.

This SMSC connection is required to utilise NowSMS. Without an SMSC connection, NowSMS is not able to send or receive messages (although very limited functionality is still available for specialised laboratory environments for phone testing).

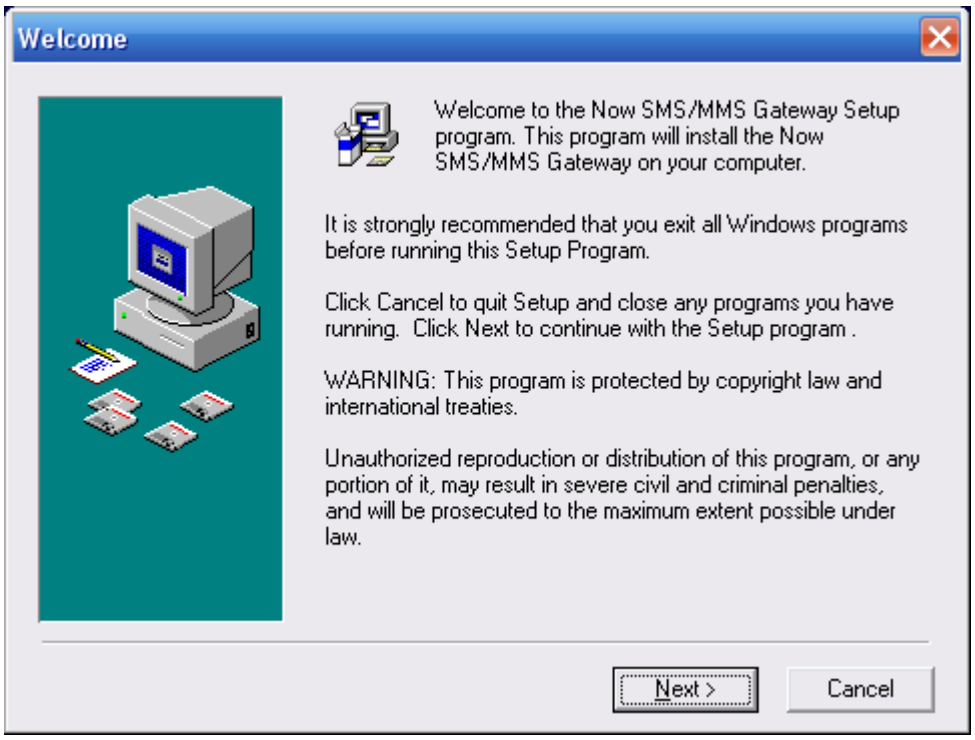
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Software Installation

The installation program for NowSMS is normally packaged in a self-installing executable program named NOWSMS.EXE. This executable might be packaged inside of a compressed ZIP-format file for electronic distribution, in which case the NOWSMS.EXE file must be extracted from the compressed ZIP file.

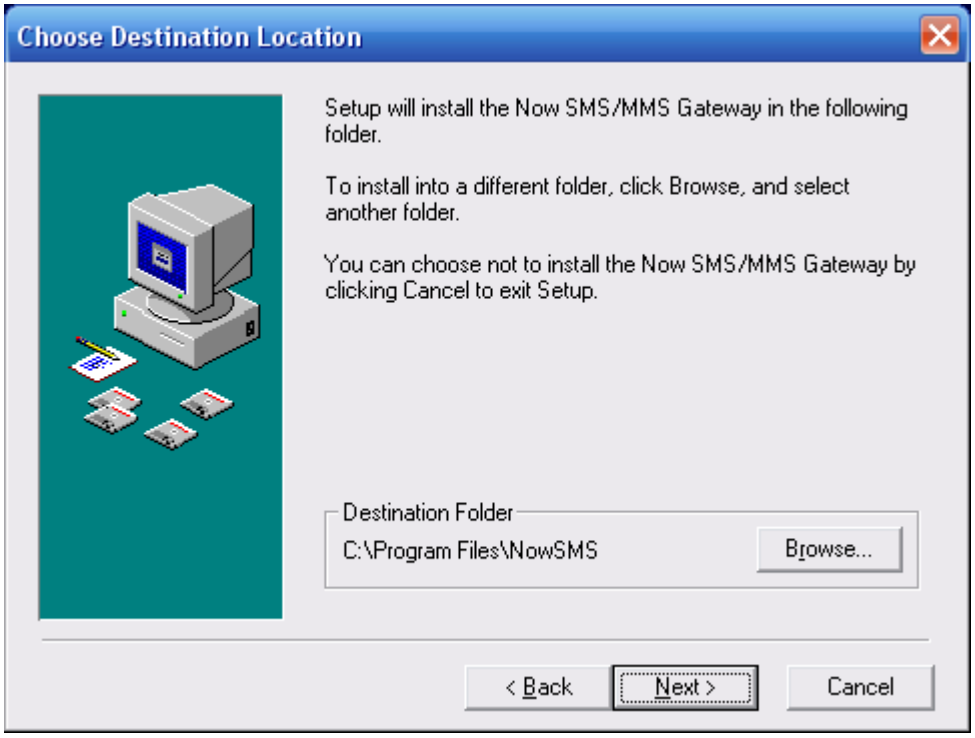
Running the NOWSMS.EXE file will begin the installation process.

An introductory screen similar to the following will be displayed:



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Next, you will be prompted for the directory in which NowSMS should be installed:



Additional informational screens may be displayed before the installation continues.

In particular, if an existing version of NowSMS is already installed, the new version of NowSMS might require that you have a NowSMS Upgrade Agreement in place. The standard practice is that each NowSMS purchase receives 12 months of free upgrades. After this 12 month period, you must purchase a NowSMS Upgrade Agreement in order to use future NowSMS upgrades.

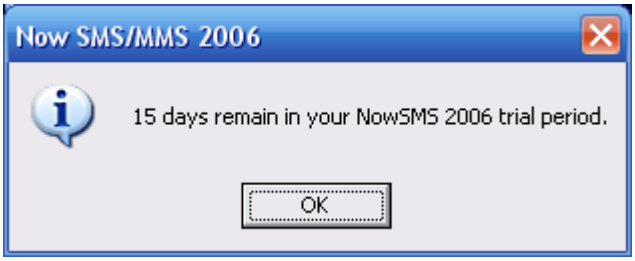
To continue the installation process, identify the type of installation that you are performing:

- New Customer: Installing Free 60-Day Trial Version *(page 11)*
- New Customer: Applying a Purchased License to an existing Trial Installation *(page 12)*
- New Customer: Installing a Purchased License Version *(page 15)*
- Existing Customer: Upgrading a NowSMS Installation *(page 18)*
- Existing Customer: Moving NowSMS to a new computer *(page21)*

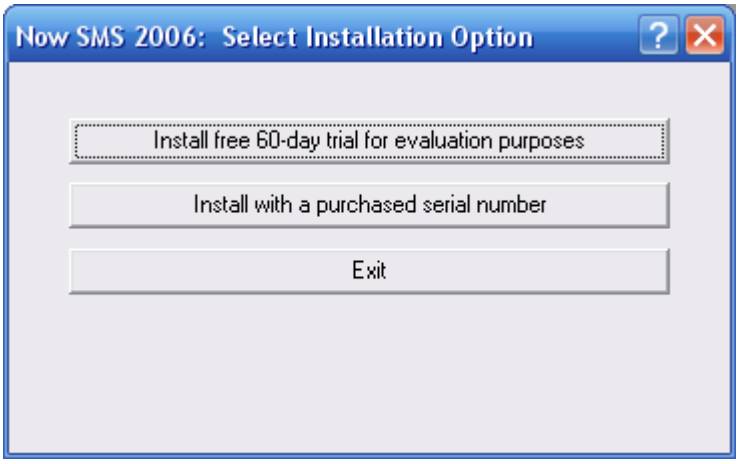
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

New Customer: Installing Free 60-Day Trial Version

When installing NowSMS 2009 for the first time, an informational message will be displayed to indicate how many days remain in the trial period:



For a first-time installation, the informational message should indicate that 60 days remain in the trial period. (NowSMS counts each day that the product is used.) You are then presented with the following options:



Select "Install free 60-day trial for evaluation purposes", and the installation will continue.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

New Customer: Applying a Purchased License to an existing Trial Installation

If you have already installed the NowSMS trial version, it is not necessary to re-install NowSMS in order to apply a purchased license to the product. However, it is possible to do this, and you may follow either the instructions in this section, or the instructions under the heading **New Customer: Installing a Purchased License Version** on page 15.

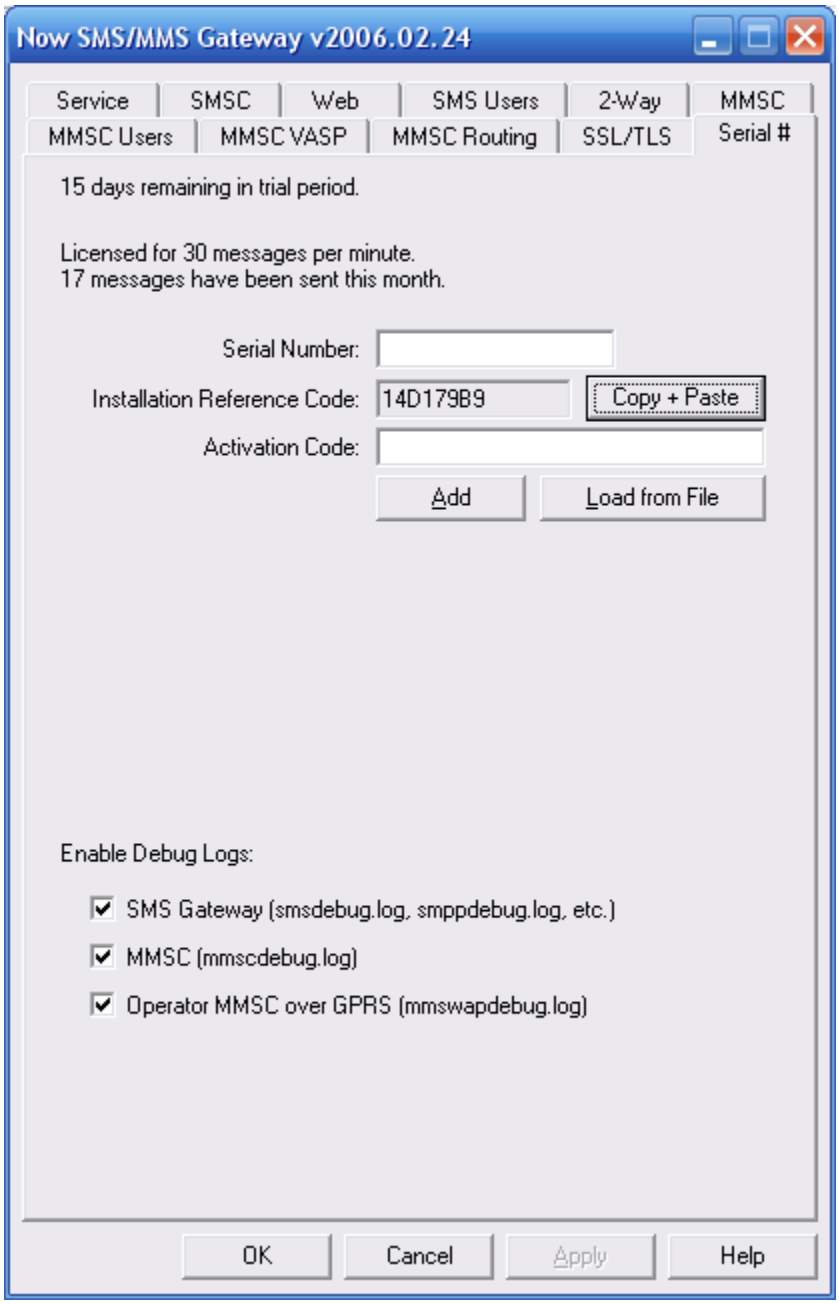
When you purchase a NowSMS license, you will not be able to apply this license to the software until you receive both a **Serial Number** and **Activation Code**.

The **Activation Code** is specific to your NowSMS installation, and will not be delivered until you first send an **Installation Reference Code** that identifies your NowSMS installation to the publisher of the NowSMS software.

The **Activation Code** is approximately 40 characters in length, and can be either entered manually, or more commonly, the publisher of the NowSMS software will send you a text file attachment in an e-mail message.

In order to receive the **Activation Code**, you must first send an **Installation Reference Code** that identifies your NowSMS installation to the publisher of the NowSMS software. This **Installation Reference Code** can be located on the **Serial #** page of the NowSMS configuration program:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



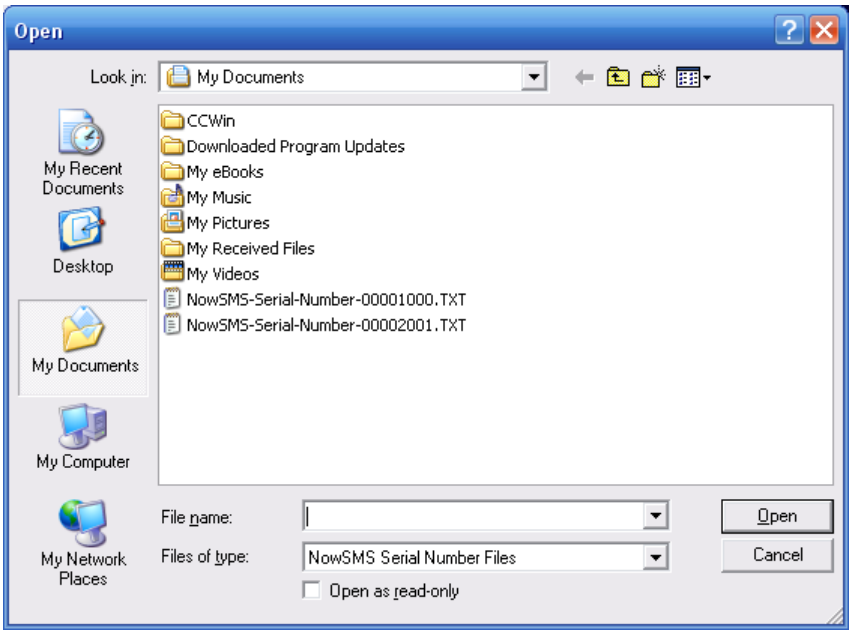
To receive your NowSMS activation code, you must supply your unique **Installation Reference Code** to Now Mobile.

You can request your activation code via e-mail or telephone. To request via e-mail, please send your serial number, company name and installation reference code to activate@nowsms.com, or to your NowSMS sales representative. To make it easier to copy and paste the code, you can use the **Copy + Paste** button to copy it into an e-mail or other document.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Once you have received your Serial Number and Activation Code, these items can be manually entered in the **Serial #** page of the NowSMS configuration.

As mentioned previously, the **Activation Code** is approximately 40 characters in length, which can be inconvenient for manual input. Normally, the publisher of the NowSMS software will send you a text file attachment in an e-mail message, which can be easily selected via the **Load from File** button.



After opening the file, the serial number will be automatically installed to complete the product licensing.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

New Customer: Installing a Purchased License Version

When you purchase a NowSMS license, you will not be able to apply this license to the software until you receive both a **Serial Number** and **Activation Code**.

The **Activation Code** is specific to your NowSMS installation, and will not be delivered until you first send an **Installation Reference Code** that identifies your NowSMS installation to the publisher of the NowSMS software.

The **Activation Code** is approximately 40 characters in length, and can be either entered manually, or more commonly, the publisher of the NowSMS software will send you a text file attachment in an e-mail message.

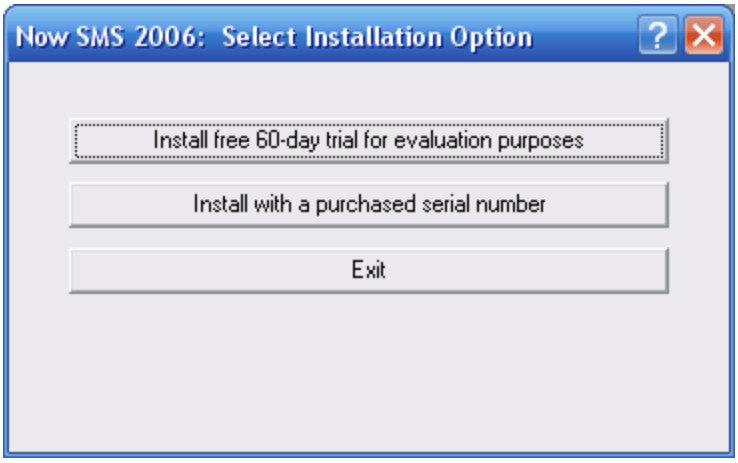
If you do not already have an **Activation Code**, you must run the NowSMS installation program to generate your **Installation Reference Code**.

As the NowSMS installation continues, an informational message will be displayed to indicate how many days remain in the trial period.

For a first-time installation, the informational message should indicate that 60 days remain in the trial period. (NowSMS counts each day that the product is used.) Select **OK** to continue.

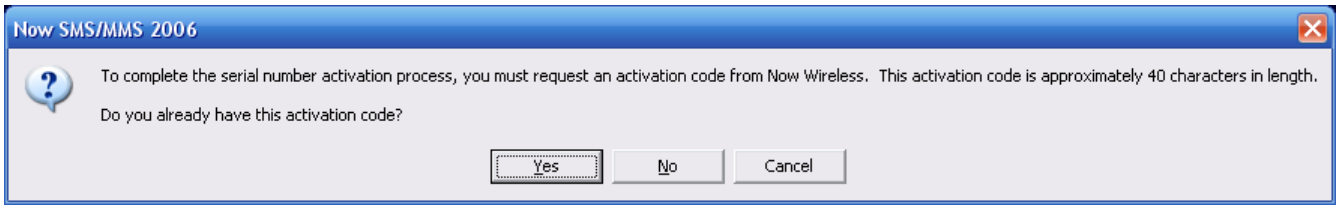
Alternatively, if the trial period has already expired, the message may indicate this and you will be asked if you have a purchased license to install. If this message is displayed, select **Yes** to continue.

You are then presented with the following options:

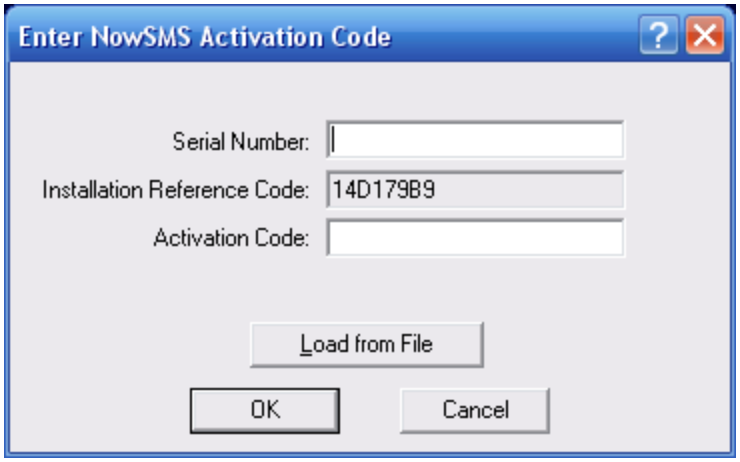


Select "Install with a purchased serial number", and the installation will continue.

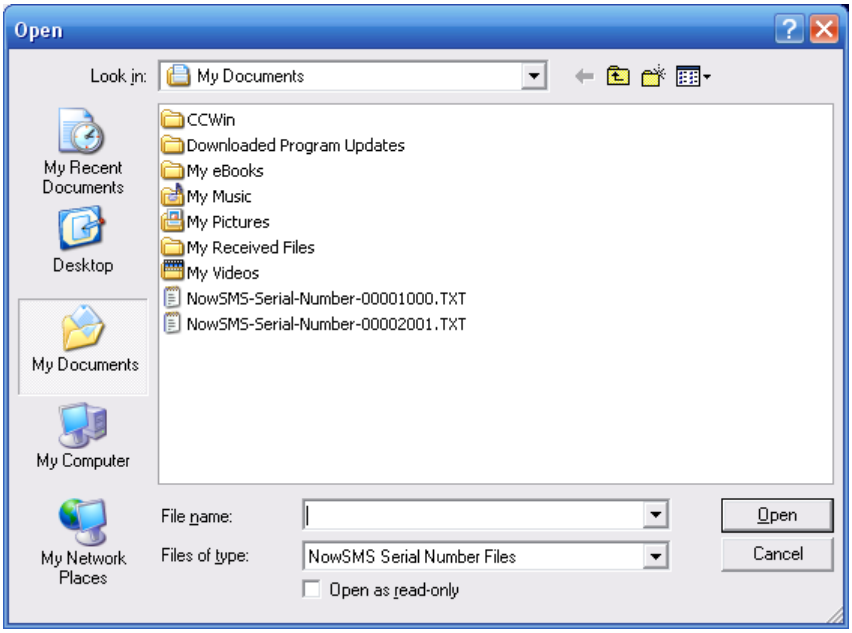
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



Selecting **Yes** will prompt for the Serial Number and Activation Code.

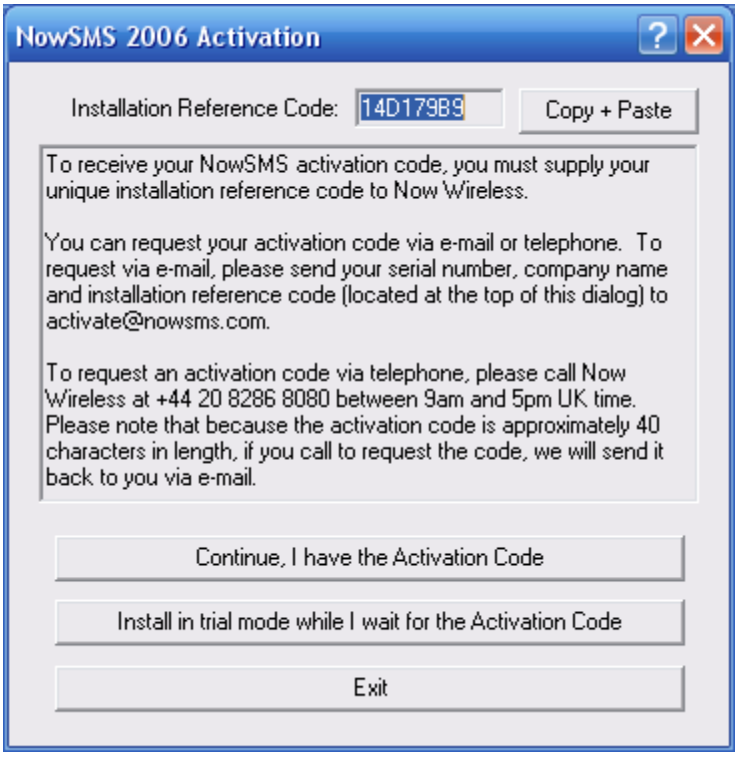


As mentioned previously, the **Activation Code** is approximately 40 characters in length, and can be either entered manually, or more commonly, the publisher of the NowSMS software will send you a text file attachment in an e-mail message, which can be easily selected via the **Load from File** button.



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

In order to receive the **Activation Code**, you must first send an **Installation Reference Code** that identifies your NowSMS installation to the publisher of the NowSMS software. If you do not yet have an Activation Code, information will be displayed regarding how to request this code:



To receive your NowSMS activation code, you must supply your unique **Installation Reference Code** to Now Mobile.

You can request your activation code via e-mail or telephone. To request via e-mail, please send your serial number, company name and installation reference code to activate@nowsms.com, or to your NowSMS sales representative. To make it easier to copy and paste the code, you can use the **Copy + Paste** button to copy it into an e-mail or other document.

It is possible to install the product in trial mode while awaiting delivery of the Activation Code. In this mode, the software will function for 60 days. Once you have received the activation code, it can be easily applied by following the instructions under the heading **New Customer: Applying a Purchased License to an existing Trial Installation** on page 12.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

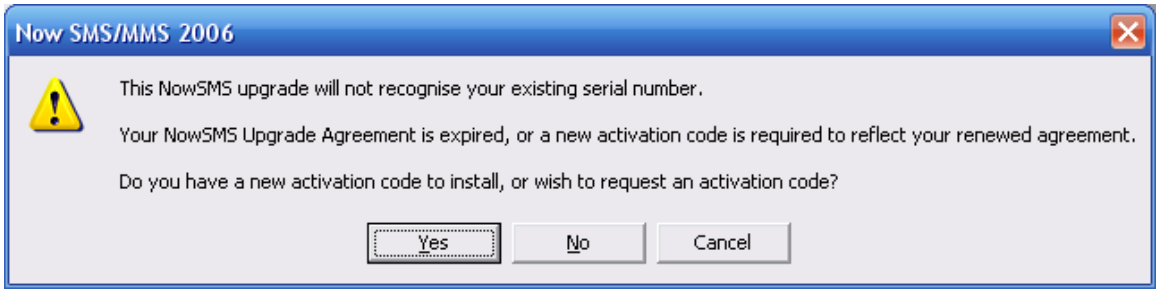
Existing Customer: Upgrading a NowSMS Installation

If you are an existing NowSMS customer without a current NowSMS Upgrade Agreement, you can continue to use existing, purchased and installed NowSMS product. You will just not be able to install upgraded versions of the NowSMS software.

If NowSMS detects an existing installation with a serial number for an older version of the product, without a current NowSMS Upgrade Agreement, it will display a warning message. If your upgrade agreement is current, no message will be displayed.

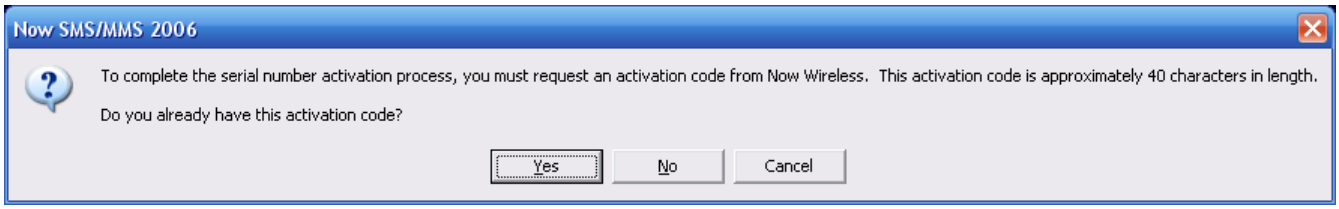
Please note that if you purchased a previous version of NowSMS within 12 months prior to the release of NowSMS 2009, your existing NowSMS activation code will allow you to install NowSMS 2009. Customers using earlier versions of NowSMS may need to purchase an upgrade, or extend their existing maintenance agreement in order to install NowSMS 2009. When upgrading, you will need to obtain an Activation Code from the publisher of NowSMS, following the instructions below.

This is the warning that might be displayed:



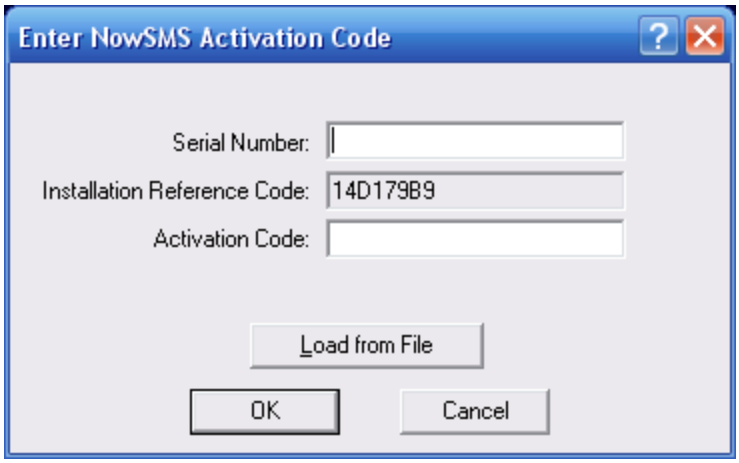
Selecting **No** or **Cancel** will abort the installation, and display contact information for the publisher of NowSMS.

Selecting **Yes** will display information on how to renew your NowSMS Upgrade Agreement and/or apply the required new **activation code**.

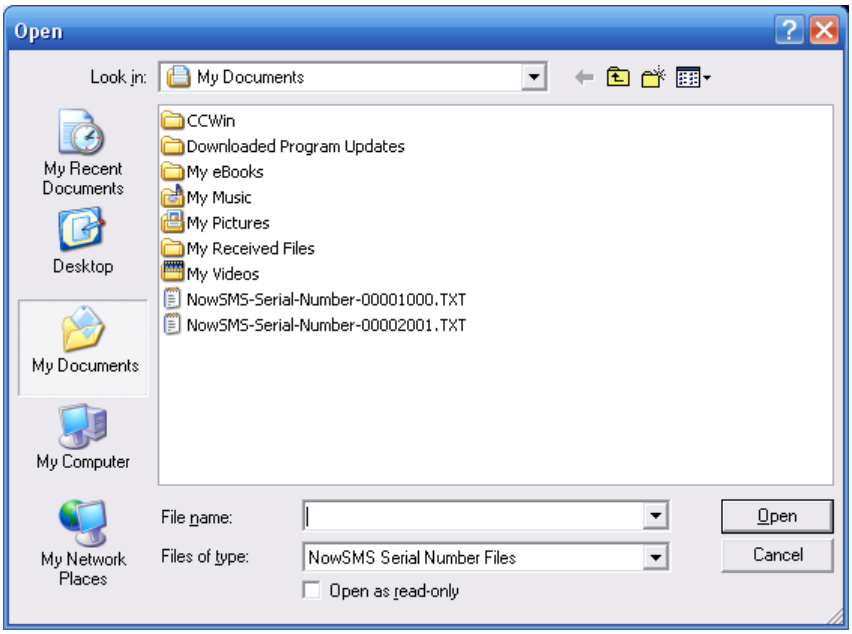


Selecting **Yes** will prompt for the Serial Number and Activation Code.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

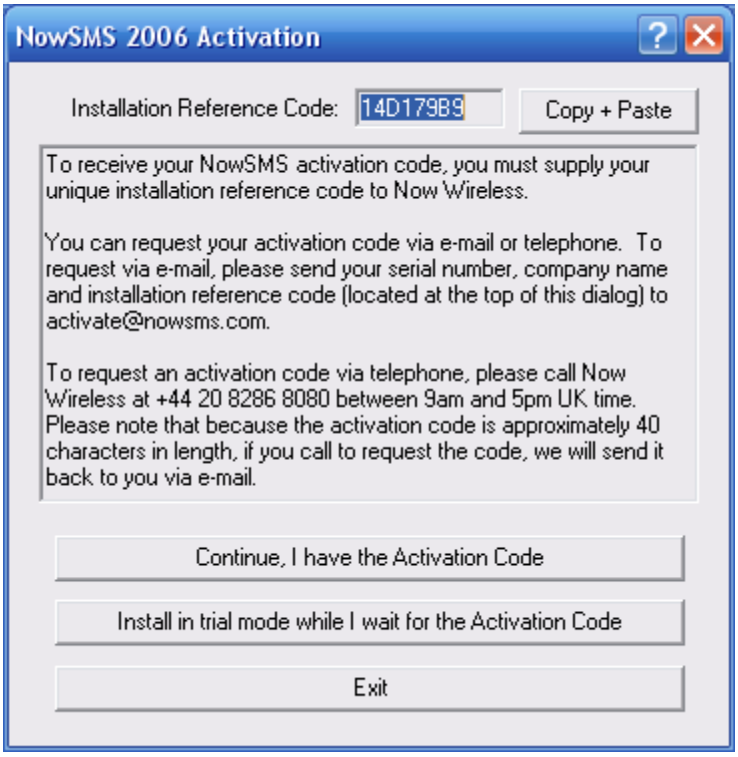


The **Activation Code** is approximately 40 characters in length, and can be either entered manually, or more commonly, the publisher of the NowSMS software will send you a text file attachment in an e-mail message, which can be easily selected via the **Load from File** button.



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

In order to receive the **Activation Code**, you must first send an **Installation Reference Code** that identifies your NowSMS installation to the publisher of the NowSMS software. If you do not yet have an Activation Code, information will be displayed regarding how to request this code:



To receive your NowSMS activation code, you must supply your unique **Installation Reference Code** to Now Mobile.

You can request your activation code via e-mail or telephone. To request via e-mail, please send your serial number, company name and installation reference code to activate@nowsms.com, or to your NowSMS sales representative. To make it easier to copy and paste the code, you can use the **Copy + Paste** button to copy it into an e-mail or other document.

It is possible to install the product in trial mode while awaiting delivery of the Activation Code. In this mode, the software will function for 60 days. Once you have received the activation code, it can be easily applied by following the instructions under the heading **New Customer: Applying a Purchased License to an existing Trial Installation** on page 12.

You can also choose to install the product in trial mode to evaluate it. If, after 60 days, you have not purchased an updated license, and you wish to continue using NowSMS, this is only possible if you restore the older version of the NowSMS software.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Existing Customer: Moving NowSMS to a new Computer

If you need to move NowSMS to a new computer, this will require a new **Activation Code**.

You can easily request a new activation code from Now Mobile.

We recommend that you follow these steps to move a NowSMS installation to a new computer:

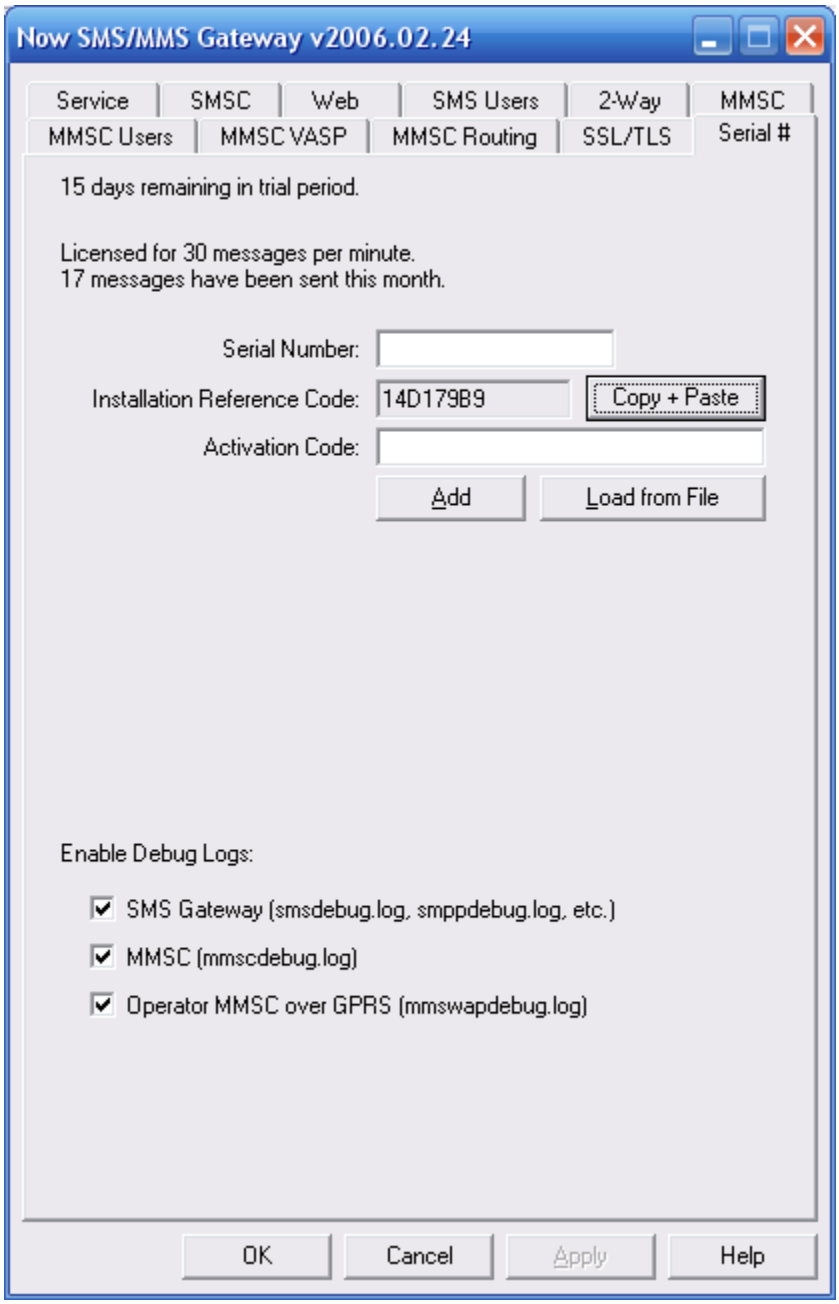
1.) Install NowSMS on the new computer, as described in the following section of this document: **New Customer: Installing Free 60-Day Trial Version** *(page 11)*.

2.) Request a new **Activation Code** from Now Mobile. The **Activation Code** is specific to your NowSMS installation, and will not be delivered until you first send an **Installation Reference Code** that identifies your NowSMS installation to the publisher of the NowSMS software.

The **Activation Code** is approximately 40 characters in length, and can be either entered manually, or more commonly, the publisher of the NowSMS software will send you a text file attachment in an e-mail message.

In order to receive the **Activation Code**, you must first send an **Installation Reference Code** that identifies your NowSMS installation to the publisher of the NowSMS software. This **Installation Reference Code** can be located on the **Serial #** page of the NowSMS configuration program:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



To receive your NowSMS activation code, you must supply your unique **Installation Reference Code** to Now Mobile.

You can request your activation code via e-mail or telephone. To request via e-mail, please send your serial number, company name and installation reference code to activate@nowsms.com, or to your NowSMS sales representative. To make it easier to copy and paste the code, you can use the **Copy + Paste** button to copy it into an e-mail or other document.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

3.) You can either wait to receive your new activation code before proceeding, or you can migrate your existing NowSMS configuration to the new computer while running NowSMS in 60-day trial evaluation mode.

To migrate your existing NowSMS configuration to the new computer, copy all files from the NowSMS directory structure of the old computer to the NowSMS directory structure of the new computer. *(It is ok to replace program files on the new computer, as this way you can ensure that you are running the same version of NowSMS on the new computer as was run on the old computer.)*

To apply our new activation code to the installation, follow the instructions described in **New Customer: Applying a Purchased License to an existing Trial Installation** *(page 12)*.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Configuring SMSC Connections

The Now SMS/MMS Gateway requires a connection to an SMSC (Short Messaging Service Centre) to interface with SMS and MMS networks. An SMSC connection can consist of one or more of the following:

GSM Modem – A GSM modem or phone connected to a PC serial port, USB port, or Bluetooth connection with an appropriate modem driver. See page 25.

SMPP (Short Message Peer to Peer Protocol) – A TCP/IP connection over the internet or a private network to a service that supports v3.3 or v3.4 of the SMPP protocol. (Note that the Now SMS/MMS Gateway also includes an SMPP server, which allows you to chain multiple gateways together.) See page 34.

UCP/EMI (Universal Computer Protocol / External Machine Interface) – A TCP/IP connection over the internet or a private network to a service that supports v3.5 or higher of the UCP/EMI protocol. UCP/EMI is primarily implemented by CMG SMSCs. See page 45.

CIMD2 (Computer Interface to Message Distribution, version 2) – A TCP/IP connection over the internet to a service that supports the CIMD2 protocol. CIMD2 is implemented by Nokia SMSCs. See page 48.

HTTP (Hyper Text Transport Protocol, e.g., the standard protocol for the "web") – A TCP/IP connection over the internet or private network to a service that accepts SMS messages via an HTTP "GET" based protocol. See page 40.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

GSM Modems

A GSM modem is a specialized type of modem which accepts a SIM card, and operates over a subscription to a mobile operator, just like a mobile phone. From the mobile operator perspective, a GSM modem looks just like a mobile phone.

For the purpose of this document, the term GSM modem is used as a generic term to refer to any modem that supports one or more of the protocols in the GSM evolutionary family, including the 2.5G technologies GPRS and EDGE, as well as the 3G technologies WCDMA, UMTS, HSDPA and HSUPA.

A GSM modem exposes an interface that allows applications such as NowSMS to send and receive messages over the modem interface. The mobile operator charges for this message sending and receiving as if it was performed directly on a mobile phone. To perform these tasks, a GSM modem must support an "extended AT command set" for sending/receiving SMS messages, as defined in the ETSI GSM 07.05 and and 3GPP TS 27.005 specifications.

GSM modems can be a quick and efficient way to get started with SMS, because a special subscription to an SMS service provider is not required. In most parts of the world, GSM modems are a cost effective solution for receiving SMS messages, because the sender is paying for the message delivery.

A GSM modem can be a dedicated modem device with a serial, USB or Bluetooth connection, such as the Falcom Samba 75 used in this document. (Other manufacturers of dedicated GSM modem devices include Wavecom, Multitech and iTegno.) To begin, insert a GSM SIM card into the modem and connect it to an available USB port on your computer.

A GSM modem could also be a standard GSM mobile phone with the appropriate cable and software driver to connect to a serial port or USB port on your computer. Any phone that supports the "extended AT command set" for sending/receiving SMS messages, as defined in ETSI GSM 07.05 and/or 3GPP TS 27.005, can be supported by the Now SMS/MMS Gateway. Note that not all mobile phones support this modem interface.

Due to some compatibility issues that can exist with mobile phones, using a dedicated GSM modem is usually preferable to a GSM mobile phone. This is more of an issue with MMS messaging, where if you wish to be able to receive inbound MMS messages with the gateway, the modem interface on most GSM phones will only allow you to send MMS messages. This is because the mobile phone automatically processes received MMS message notifications without forwarding them via the modem interface.

It should also be noted that not all phones support the modem interface for sending and receiving SMS messages. In particular, most smart phones, including Blackberries, iPhone, and Windows Mobile devices, do not support this GSM modem interface for sending and receiving SMS messages at all at all. Additionally, Nokia phones that use the S60 (Series 60) interface, which is Symbian based, only support sending SMS messages via the modem interface, and do not support receiving SMS via the modem interface.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

When you install your GSM modem, or connect your GSM mobile phone to the computer, be sure to install the appropriate Windows modem driver from the device manufacturer. To simplify configuration, the Now SMS/MMS Gateway will communicate with the device via this driver. If a Windows driver is not available for your modem, you can use either the "Standard" or "Generic" 33600 bps modem driver that is built into windows. A benefit of utilizing a Windows modem driver is that you can use Windows diagnostics to ensure that the modem is communicating properly with the computer.

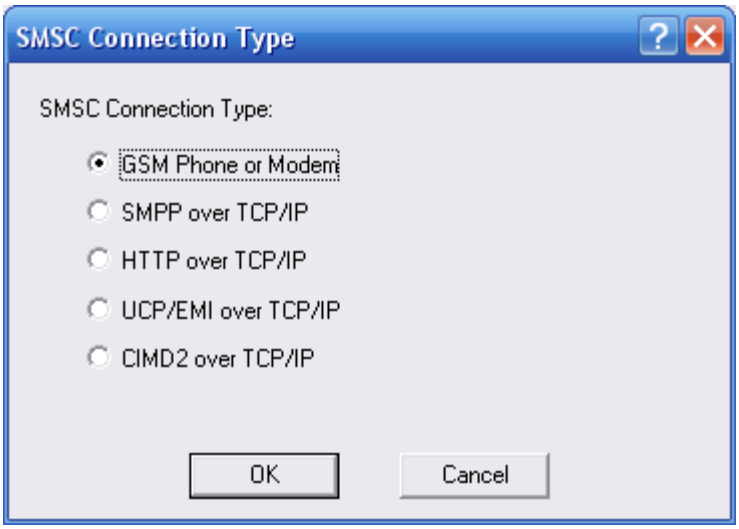
The Now SMS/MMS gateway can simultaneously support multiple modems, provided that your computer hardware has the available communications port resources.

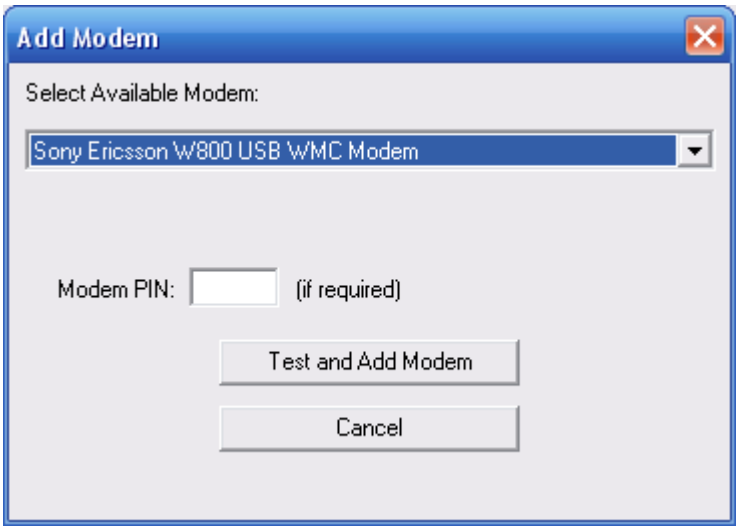
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

To define which modems are to be utilized by the gateway, select the "**SMSC**" configuration dialog from the gateway configuration dialog:

If no modems are yet to be defined, only the "**Add**" button will be available on this dialog. Select "**Add**", and then "**GSM Phone or Modem**" to display a list of available modem drivers on your computer.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved





Select an available modem and press the "**Test and Add Modem**" button. The gateway will then attempt to initialize the modem, and confirm that the modem supports the necessary interfaces to send and receive SMS messages. The modem will only be added to the configuration if the gateway confirms that it can properly communicate with the modem. If the SIM card for the modem requires a 4-digit PIN, please supply it in the field provided.

NowSMS will verify that it can communicate with the modem, and that the modem supports several of the basic AT commands defined in the ETSI GSM 07.05 (3GPP TS 23.005) specification. If the modem is successfully tested, NowSMS will display the following:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If there is a problem testing the modem, NowSMS will display an error message.

GSM Modem Troubleshooting Tips

When you encounter any error initialising a GSM modem, we recommend the following troubleshooting steps outlined below. These general troubleshooting steps will be followed by suggestions that are specific to particular error conditions.

- ✔ It is possible to define a modem connection by either selecting the name of the Windows modem driver, or selecting the name of the COM port to which the modem is connected. If you are attempting to connect to a modem by specifying a COM port and are experiencing problems, the best way to troubleshoot this problem is to ensure that you are connecting to the phone or modem using a Windows modem driver supplied by the phone or modem manufacturer, instead of going directly to a COM port. When you configure a GSM modem connection for the gateway, you are presented with a list of modem drivers installed on your system, as well as a list of COM ports. You will achieve better results going through the modem driver. If you have not installed a Windows modem driver for your device, visit the manufacturer web site, or use the CD supplied by the manufacturer, and install the appropriate modem driver. If the manufacturer does not supply a Windows modem driver (such as Wavecom), we recommend you manually define the "Standard 33600 bps Modem" driver for the modem.
- ✔ Assuming an appropriate modem driver is installed, go into the Windows Control Panel, and select the "Modems" or "Phone and Modem Options" applet. In the diagnostics section, ensure that you are able to use "Query Modem" to interface with your modem, which will ensure that Windows is able to properly communicate with the modem. The Now SMS/MMS gateway will not be able to access the modem if it is not accessible to Windows. If Windows indicates that another application is already using the modem, then you need to determine which application is involved. If you are using a phone as a modem, there may be a communications suite associated with the modem that opens a connection to the modem for phone book maintenance or other functions, which needs to be disabled. For other types of errors, follow the instructions from the device manufacturer if you encounter problems accessing the modem in the Windows Control Panel.
- ✔ Another common problem is an incorrect cable, or a faulty cable. Some phone manufacturers have different cables for different purposes. You want to the type of cable that allows your PC to use the phone as a modem (sometimes referred to as a "data/fax cable"). For example, most older Nokia phones have DLR-3 and DAU-9 types of cables. The DLR-3 type is for data/fax applications, and the DAU-9 is for use with Logo Manager. The Now SMS/MMS gateway requires the DLR-3 type (data/fax). For newer Nokia devices, as well as devices for other manufacturers, verify that you have the correct cable for using the device as a data modem.
- ✔ Is there a PIN associated with the SIM in the modem? Try putting the SIM into a mobile phone and see if it prompts for a PIN. If it does, try removing the PIN and trying again. (NowSMS supports modem PINs, but some modem drivers may have PIN related problems.)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

- ✔ Try turning off the power to the modem and restarting the modem. If the problem reoccurs, and a power cycle of the modem consistently resolves the problem, this suggests that the modem was in a hung state which might potentially be resolved by a firmware upgrade from the modem manufacturer.

- ✔ Try rebooting the PC. If the problem reoccurs, and a PC reboot consistently resolves the problem, this suggests that the software modem driver was in a hung state which might be potentially resolved by an upgrade of the Windows modem driver software from the modem manufacturer.

- ✔ Try de-installing and re-installing the Windows modem driver associated with the modem.

Additional information is supplied below regarding specific modem initialisation errors that may be returned by NowSMS:

Unable to access modem at COMx: -- Error 5 -- Access Denied -- Another application is already using this device: This error message indicates that another Windows application is already communicating with the modem, and only one application can communicate with the modem at a time. It is possible that some software that was installed with your modem may be automatically opening a connection to the modem for its own purposes, so we recommend that you try disabling some of the more advanced features of any communications suite software that came with your phone or modem. If the error persists, try connecting the modem to a different port. We also recommend that you attempt further diagnostics within the Windows Control Panel, using the "Query Modem" function under the Diagnostics section of the Phone & Modem Options applet.

Unable to access modem at COMx: -- Error xxxx -- yyyyyyyyyy: This error message indicates that Windows is reporting a problem accessing the communications port associated with the modem. In this case, COMx indicates the port number associated with the modem. xxxx indicates the Windows error number. yyyyyyyyyy is descriptive text for the error as provided by Windows. If the problem condition is not obvious based upon the supplied error information, we suggest querying the NowSMS discussion board (http://www.nowsms.com/messages) for potential information. If the error persists, try connecting the modem to a different port. We also recommend that you attempt further diagnostics within the Windows Control Panel, using the "Query Modem" function under the Diagnostics section of the Phone & Modem Options applet.

Unable to initialize modem: Error XXXXXXXX from lineOpen: This error message indicates that the Windows Telephony API (TAPI) subsystem could not open a connection to the modem. In most cases, this is the same as the "Error 5 -- Access Denied" error above, indicating that another Windows application is already communicating with the modem. We suggest following the same suggestions as offered above. In some cases it may be necessary to de-install the Windows modem driver, and re-install it.

Unable to initialize modem: Error XXXXXXXX from lineGetID: This error message indicates that Windows could not get a response back from the modem, when it attempted to communicate with the modem. We recommend that you attempt further diagnostics within the Windows Control Panel, using the "Query Modem" function under the Diagnostics section of the Phone & Modem Options applet. If the problem persists, try turning off the power to the modem and restarting it. If a power cycle of the modem resolves the problem, this suggests that the modem was in a hung state which might potentially be resolved by a firmware upgrade from the modem manufacturer. If the problem persists,

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

try rebooting the PC. If a PC reboot resolves the problem, then this suggests that the software modem driver was in a hung state which might be potentially resolved by an upgrade of the Windows modem driver software from the modem manufacturer. If the above suggestions do not resolve the problem, we recommend that you attempt to de-install the Windows modem driver, and then re-install it.

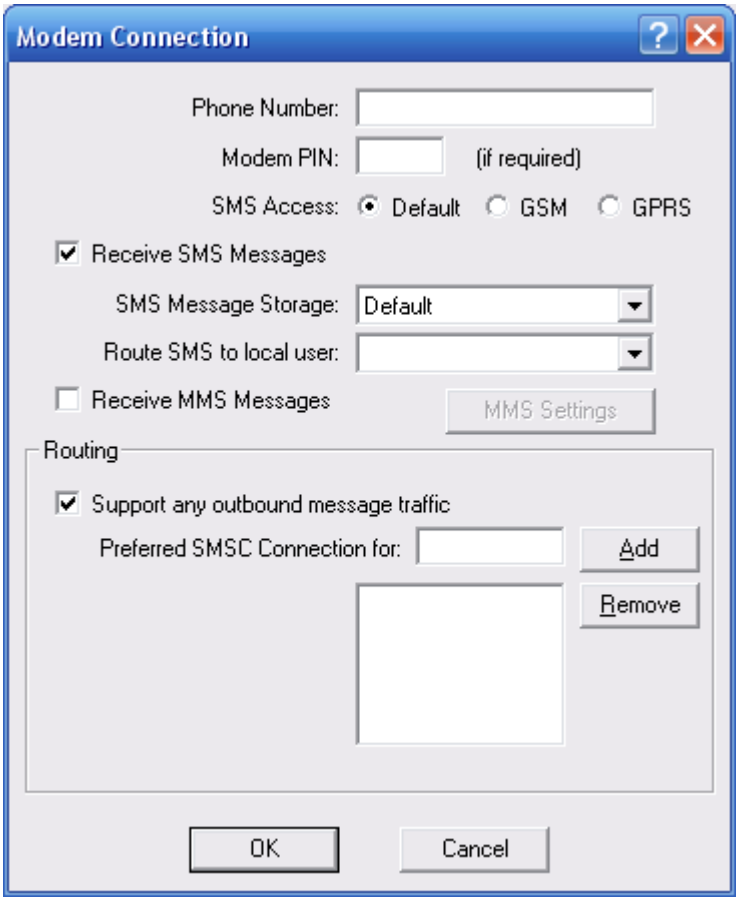
Modem does not support SMS -- ERROR: This error message indicates that the modem does not support some of the required commands as defined in ETSI GSM 07.05 (3GPP TS 23.005). Specifically it is rejecting the AT+CSMS=0 command. It may be possible that you have selected the wrong modem (for example an internal modem built into the PC), or that the modem does not support the AT commands for sending/receiving SMS as defined in the above referenced specification. Some phones, such as the SonyEricsson P800/P900/P910 do provide the ability to function as a GPRS modem for internet connectivity, but they do not support the SMS-related AT commands. You may want to query the NowSMS Discussion Board (http://www.nowsms.com/messages) for more information regarding your phone or modem model.

Modem does not support SMS text or PDU mode commands - ERROR: This error message indicates that the modem does not support some of the required commands as defined in ETSI GSM 07.05 (3GPP TS 23.005). Specifically, it is rejecting both the AT+CMGF=0 and AT+CMGF=1 commands, where NowSMS is trying to determine if the modem can support either text or PDU (binary) mode. It is very unusual for this error to be returned, therefore you may want to query the NowSMS Discussion Board (http://www.nowsms.com/messages) for more information regarding your phone or modem model.

Unable to access modem, ensure that it is powered on and passes diagnostic tests: This error message is displayed when there is a communications error communicating with the modem. Another error message should have been displayed prior to this message, and that error message contains more specific information about the nature of the problem.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

After the Modem is added, there are additional Properties that can be configured for the modem connection. Highlight the modem name in the "**SMSC**" list, and press the "**Properties**" button:



The "**Phone Number**" field is used to tell NowSMS the phone number associated with this modem. It is not mandatory to supply a phone number, however if a phone number is specified, it will be possible to submit messages to the gateway in such a way that if multiple SMSC or GSM modem connections are defined, the message will be sent out via this connection. (This is possible by including a "Sender" parameter in a URL request to submit a message, where the value of this parameter matches the "Phone Number" field configured for a specific GSM modem. For more information, refer to **Submitting SMS Messages - URL Parameters** on page 187.) When receiving SMS or MMS messages, the phone number will be supplied as the receiving address of the message, allowing applications to determine which modem received the message in an installation with multiple GSM modem connections.

A Modem PIN is a 4-digit code that is used for security purposes. If a PIN is configured on the SIM card installed in the modem, the phone or modem cannot be used until the PIN is entered. If you want the gateway to automatically supply the PIN to the modem upon startup, supply this PIN in the "**Modem PIN**" field.

The "**SMS Access**" setting specifies whether SMS messages should be sent by the modem over the circuit-switched or packet-switched network. This setting is not limited to GSM

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

and GPRS environments, but also applies to EDGE and 3G/WCDMA/UMTS. Setting the value to "**Default**" uses the default as configured on the modem. Setting the value to "**GSM**" tells the modem to use the circuit-switched network for sending SMS. Setting the value to "**GPRS**" tells the modem to use the packet-switched network. Generally speaking, the packet-switched network will offer better performance, however it is not supported by all operators, in which case the "**GSM**" setting must be used. Similarly a modem might default to SMS over the packet-switched network, and if you experience a problem sending SMS with a particular modem it might be necessary to manually configure the "**GSM**" setting to tell the modem to use the circuit-switched network instead.

If the Now SMS/MMS Gateway should process SMS messages received by the attached modem, the "**Receive SMS Messages**" setting should be enabled. The "**SMS Message Storage**" location should be left at "Default" unless otherwise instructed by technical support. For more information on how to process received SMS messages, please refer to **2-Way SMS Support** on page 236.

It is possible to configure NowSMS to route all SMS messages received via this modem to a user account on the NowSMS server by selecting the user account in the "**Route SMS to local user**" setting. It is possible for a user to connect to the NowSMS server either using SMPP (an SMS specific protocol) or POP3 (an e-mail protocol). These user accounts are defined on the "**SMS Users**" page of the NowSMS configuration dialog *(see page 64)*.

If the Now SMS/MMS Gateway should process MMS messages received by the attached modem, the "**Receive MMS Messages**" setting should be enabled. The "**MMS Settings**" dialog will display a dialog with additional configuration settings that are required for enabling the gateway to be able to receive MMS messages from the operator network. Please note that a dedicated GSM modem device (not a phone acting as a modem) is required in most instances in order to support the receiving of MMS messages, and in most cases the modem device must also support GPRS. For additional configuration information, please refer to **Connecting to an Operator MMSC - Using a GPRS Modem** on page 136. For additional configuration information on processing received MMS messages, please refer to **2-Way MMS Support** on page 240.

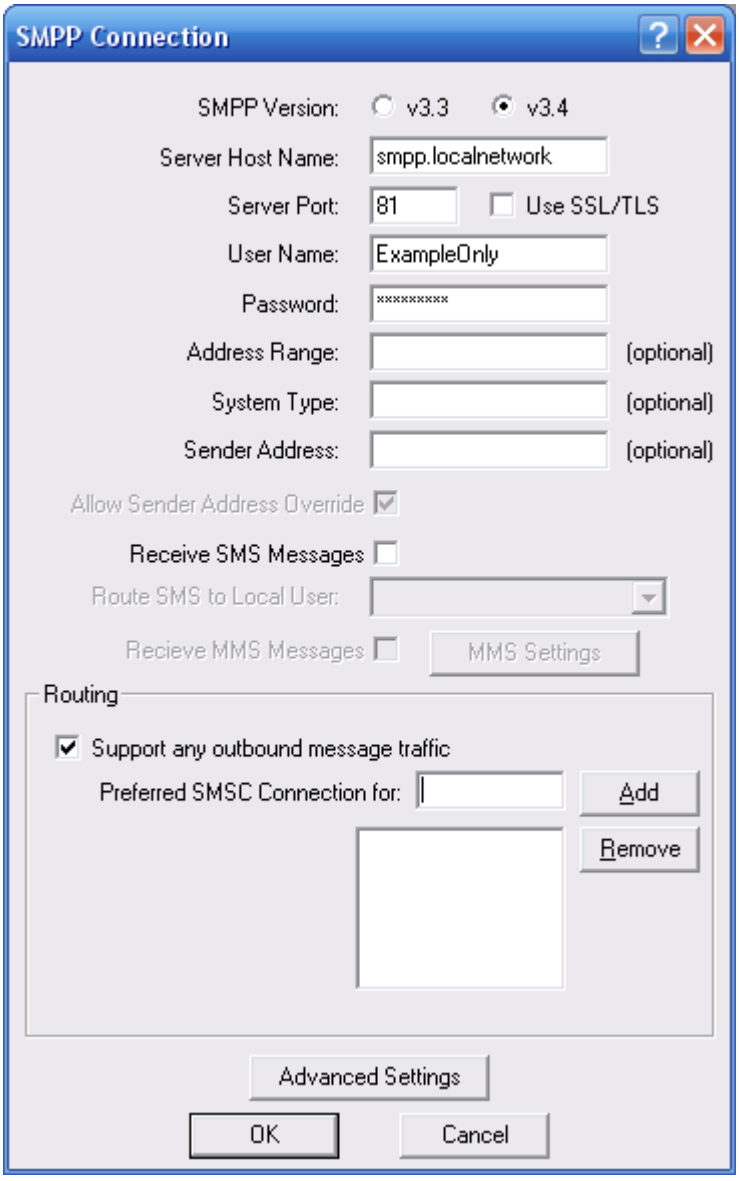
The "**Routing**" group of options is used when multiple SMSC connections are defined to the gateway. These options define what messages should be routed to this connection. The Routing options are common to SMPP, GSM Modem, UCP/EMI and HTTP connections, and are described in the **Routing Options** section *(page 52)*.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

SMPP SMSC

The gateway supports the SMPP (Short Message Peer to Peer) protocol, version 3.3 or 3.4, to connect to an SMSC over the internet or other private TCP/IP network. As SMPP is designed and optimised specifically for SMS transmission, many mobile operators provide SMPP connections for higher volumes of SMS traffic.

To add an SMPP connection, select "**Add**" from the "**SMSC**" configuration dialog. Then select "**SMPP over TCP/IP**".



"**SMPP Version**" specifies the version of the SMPP protocol to use. The gateway supports "v3.3" and "v3.4".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"**Server Host Name**" specifies the TCP/IP address or host name of the SMPP server.

"**Port**" specifies the TCP/IP port on the SMPP server to which the gateway should connect.

"Use SSL/TLS" should only be checked if the service provider specifically indicates that they support SSL or TLS encryption for the SMPP connection.

"**User Name**" specifies the user name (sometimes called System ID) for the gateway to use when connecting to the SMPP server.

"**Password**" specifies the password for the gateway to use when connecting to the SMPP server.

"**Address Range**" is a parameter used primarily when receiving messages. Set this field only if instructed to do so by your SMPP service provider.

"System Type" is an optional login parameter that should be set only if required by the SMPP server. The SMPP system administrator will provide this value, which when required, is usually a short text string.

"Sender Address" specifies the default sender address (phone number) to apply to outbound SMS messages. The SMPP server may override this setting.

Check **"Allow Sender Address Override"** if you want to allow messages submitted to the gateway to be able to specify a sender address. If this box is checked, and a sender address is present in a message being submitted to an SMPP based SMSC, the sender address in the message will be submitted to the SMSC. The SMPP server may override this setting.

Check the "**Receive SMS Messages**" box if you wish to receive messages from the SMPP server. When this box is checked, by default, the gateway will connect to the SMPP server with two separate connections, one bound as a transmitter and the other bound as a receiver. Some SMS providers prefer that only a single connection resource is used, bound as a transceiver. It is possible to configure a single transceiver connection to be used on the **"Advanced Settings"** page.

It is possible to configure NowSMS to route all SMS messages received via this SMSC connection to a user account on the NowSMS server by selecting the user account in the **"Route SMS to local user"** setting. It is possible for a user to connect to the NowSMS server either using SMPP (an SMS specific protocol) or POP3 (an e-mail protocol). These user accounts are defined on the **"SMS Users"** page of the NowSMS configuration dialog *(see page 64)*.

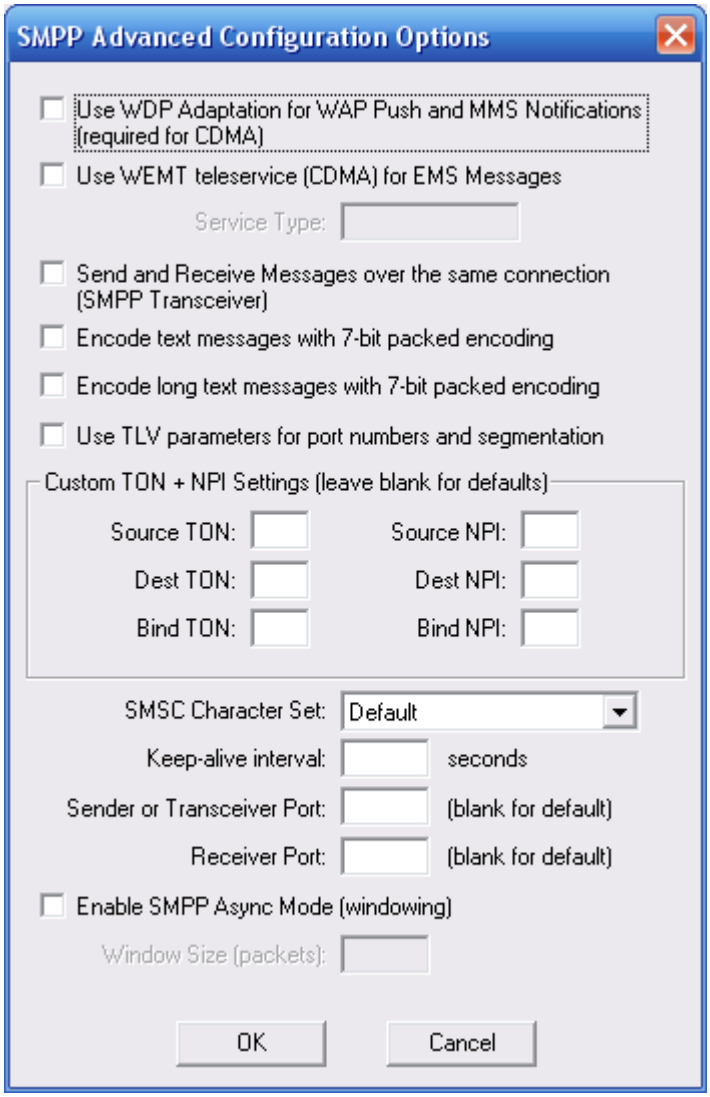
Check the **"Receive MMS Messages"** box if you will be receiving MMS Notification messages via the SMSC. Note that ***most service providers do not support*** the routing of MMS notification messages via an SMSC connection. If MMS messages will be received via the SMSC connection, it is necessary to also configure additional MMS settings. The **"MMS Settings"** dialog will display a dialog with additional configuration settings that are required for enabling the gateway to be able to receive MMS messages from the operator network. For additional configuration information, please refer to **Connecting to an Operator MMSC – Using a GPRS Modem** on page 136. For additional configuration information on processing received MMS messages, please refer to **2-Way MMS Support** on page 240.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The "**Routing**" group of options is used when multiple SMSC connections are defined to the gateway. These options define what messages should be routed to this connection. The Routing options are common to SMPP, GSM Modem, UCP/EMI and HTTP connections, and are described in the **Routing Options** section *(page 52)*.

When the "**OK**" button is selected, the gateway will attempt to connect to the SMPP server to verify the configuration information provided. Diagnostic information will be displayed if the connection fails. The connection will only be added to the configuration after a successful connection to the SMPP server.

Many SMPP SMSC providers will also tell you to that you need to configure certain additional parameters in your SMPP software. *(Editor's Note: Or many won't tell you, and you have to make educated guesses.)* NowSMS provides the ability to configure these settings on the **SMPP Advanced Configuration Options** dialog, accessible by pressing the **Advanced Settings** button when defining properties for an SMPP connection.



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Use WDP Adaptation for WAP Push and MMS Notifications (required for CDMA) – This setting is primarily used when connecting to a CDMA based SMSC. By default, NowSMS generates WAP Push messages (and MMS Notification messages which are based upon WAP Push) using a format that is specific to GSM (and WCDMA/UMTS) environments. When this configuration option is enabled, NowSMS uses a protocol independent format known as "WDP Adaptation". This is usually the only practical option for delivering WAP Push messages in a CDMA environment. In this case, NowSMS will submit messages to the SMPP server using the "WAP" teleservice (SMPP service type). For more information on this option, please refer to the NowSMS Technical Bulletin titled "**Using NowSMS as an MMSC in CDMA or CDMA2000 Environments**" on page 379.

Use WEMT teleservice (CDMA) for EMS Messages – This setting is primarily used when connecting to a CDMA based SMSC. When NowSMS detects that a message is of the EMS format, it will submit the message to the SMPP server using the specified **Service Type** value *(usually WEMT)*.

Send and Receive Messages over the same connection (SMPP Transceiver) - By default, when NowSMS is configured to both send and receive messages via an SMPP connection, NowSMS will connect to the SMPP server with two separate connections, one bound as a transmitter and the other bound as a receiver. Some SMS providers prefer that only a single connection resource is used, bound as a transceiver. When this option is enabled, NowSMS will use a single transceiver connection instead of two separate connections.

Encode text messages with 7-bit packed encoding – In GSM environments, when messages are actually delivered to a handset, simple text messages are compacted into a 7-bit packed encoding. *(This allows 160 text characters to fit into 140 binary bytes of data.)* Normally, in SMPP environments, this compaction is performed by the SMSC. However, if you receive garbage text messages when sending via NowSMS, try enabling this setting which will cause NowSMS to perform the compaction when submitting via the SMPP server. (This is normally only required for SMPP 3.3 servers.)

Encode long text messages with 7-bit packed encoding – Similar to the above setting, this setting applies only for long messages (and potentially other messages such as EMS, which include UDH elements). Some SMSCs expect normal text messages to be encoded use regular text encoding. However, if the message includes any UDH elements, the SMSC treats it as a binary message and assumes that any 7-bit packing of the text has already been performed. If you receive garbage messages when sending longer text messages via NowSMS, try enabling this setting.

Use TLV parameters for port numbers and segmentation – By default, when sending long text messages which must be segmented over multiple SMS messages, or when sending messages that include port numbers such as for WAP push, NowSMS automatically generates the appropriate GSM User Data Header (UDH) fields. When this option is enabled, NowSMS will instead use optional TLV parameters in the SMPP header for port numbers and segmentation. Specifically port numbers will be encoded in the source_port and destination_port parameters, and segmentation will be encoded in the sar_msg_ref_num, sar_total_segments, and sar_segment_seqnum parameters.

Custom TON + NPI Settings

Your SMS provider might be tell you to specify particular Bind TON, Bind NPI, Source TON, Source NPI, Destination TON or Destination NPI values. (TON = Type Of Number, NPI = Numbering Plan Indicator)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The Now SMS/MMS Gateway uses intelligent defaults for the TON and NPI values which are sufficient for 99% of SMPP connections.

The Source TON and NPI are settings that apply to the sender address that is associated with messages that are submitted by the Now SMS/MMS Gateway to the SMSC.

NowSMS automatically sets the Source TON to "1" if the sender address is in international format (starts with a "+" character), and the Source NPI to "1".

If the sender address is not in international format, NowSMS checks the to see if the address contains alphabetic characters. If it does, then it is considered to be an alphanumeric sender, and the Source TON is automatically set to "5", with Source NPI" set to "0".

Next NowSMS checks to see if the address is 5 digits or less, if it is, then it is considered to be a short code, and the Source TON is automatically set to "3", and Source NPI to "0". (To change the default max short code length, or to disable this check, edit SMSGW.INI, and add MaxSMPPShortCodeLen=# under the [SMSGW] header. Addresses that are this number of digits or less are considered to be short codes.)

If the above checks are still not satisfied, NowSMS automatically sets the Source TON to "0", and the Source NPI is set to "1".

The Destination TON and NPI are settings that apply to the recipient addresses for messages that are submitted by the Now SMS/MMS Gateway to the SMSC. NowSMS automatically sets the Destination TON to "1" if the recipient address is in international format (starts with a "+" character). If the recipient address is not in international format, NowSMS automatically sets the Destination TON to "0". In both cases, the Source NPI is set to "1".

If it is necessary to adjust these TON and NPI settings, they can be adjusted via the **Custom TON + NPI Settings** section.

If a TON or NPI value is explicitly set in the SMSGW.INI file, this value will be used in place of the automatic determination described above.

The Bind TON and Bind NPI settings are used when binding to the SMSC only. By default, NowSMS will use 0 for both values.

SMSC Character Set – By default, NowSMS uses the GSM character set when submitting SMS messages via SMPP, and it indicates via the data_coding parameter that the default encoding is used. If you experience a problem where @ symbols and other characters do not appear correctly in messages, your SMSC might be expecting a different character set. Assuming that you are working in a GSM environment primarily, first try changing this setting to "IA5 (GSM)". When this setting is applied, NowSMS will still use the GSM character set, but it will set a flag in the header to indicate this. If the character problems persist, change this setting to "iso-8859-1 (Latin)", which is the standard character set used in Western Europe.

Keep-alive Interval – This setting specifies a value, in number of seconds, between which NowSMS will automatically send enquire_link commands to the SMSC. Most SMSC require that connected clients periodically send an enquire_link command to verify that they are

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

still connected and functioning correctly. If this parameter is not specified, a default of 58 seconds is used. A value of 0 seconds can be used to disable NowSMS from sending any enquire_link commands.

Sender or Transceiver Port – Some SMSCs have firewall restrictions which will only allow you to connect to the SMSC from a specific IP address and a specific port number. Normally, NowSMS will use a dynamically allocated port number when connecting to an SMSC. However, this setting allows the port number to be fixed. This setting applies to the port number to be used for the connection that uses the transmitter (send) or transceiver SMPP bind. (Note: If you use fixed port numbers, after disconnecting from an SMSC, it may take up to 5 minutes before you are allowed to reconnect. This is a TCP/IP limitation on reconnects after a disconnect when the same source and destination ports are used for the reconnection attempt.)

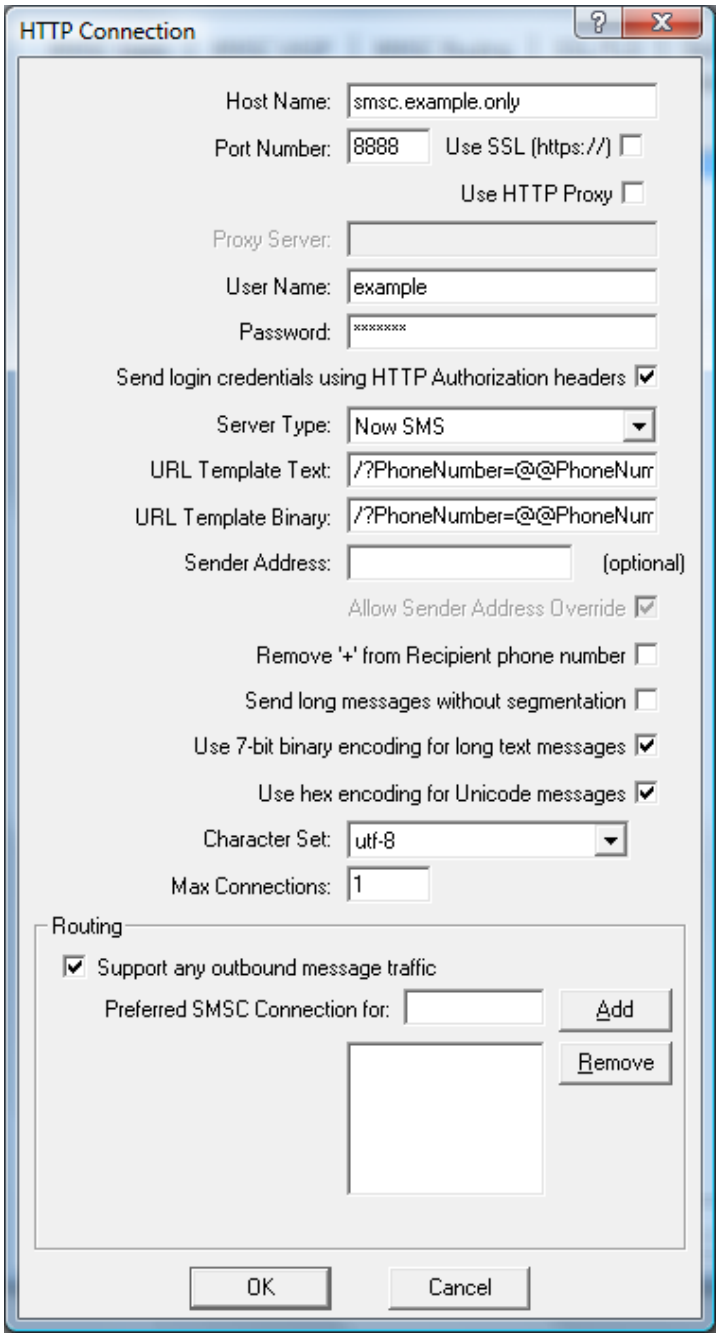
Receiver Port -Similar to the "Sender or Transceiver Port" setting, this fixed port number is applied to the connection that is used for the receiver SMPP bind.

Enable SMPP Async Mode (windowing) – Enabling this setting allows messages to be sent more rapidly over an SMPP connection. The typical SMPP message flow involves the sender submitting a message to the server, and the receiver sending back a response to acknowledge receipt of the message. When async mode is not enabled, the sender does not send the next message until the previous acknowledgment has been received. Depending on the speed (and latency) of the connection between the sender and the receiver, and the amount of processing that the receiver performs before sending back an acknowledgment, this can have a serious impact on the overall connection performance. Enabling SMPP Async Mode tells NowSMS that it does not need to wait for an acknowledgment for the previous message before submitting the next message. Instead, NowSMS will send up to the defined **Window Size** number of messages without receiving an acknowledgment. As long as this is properly supported by the SMS service provider, this can provide a great increase in potential throughput, and is generally required for any single SMPP connection to exceed 3 messages per second.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

HTTP SMSC

The gateway supports the HTTP (Hyper Text Transport Protocol) protocol to connect to an SMSC over the internet or other private TCP/IP network. Please note that this functionality is for connecting to an SMS service provider that accepts SMS messages via HTTP. (Trying to connect back to your own NowSMS server, or to your regular web site will not cause any messages to actually be sent.) To add an HTTP connection, select "**Add**" from the "**SMSC**" configuration dialog. Then select "**HTTP over TCP/IP**".



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"**Host Name**" specifies a DNS host name or TCP/IP address for connecting to the service.

"**Port Number**" specifies the HTTP port number to use when connecting to the service.

Check **"Use SSL (https://)"** if the URL for connecting to the service provider is https:// based instead of http://.

Check **"Use HTTP Proxy"** if the gateway must connect to the HTTP server via a proxy server, and supply the host name or TCP/IP address and port number of the proxy server in the "**Proxy Server**" field using a format of "host.name:9999", where "host.name" is the DNS host name or TCP/IP address of the proxy server, and "9999" is the port number of the proxy server.

"**User name**" and "**Password**" specify a user account and password to use when connecting to the service.

"**Send login credentials using HTTP Authorization headers**" refers to how the user name and password information is sent to the HTTP server. The login information can either be sent as replaceable parameters in the URL request, or using the "HTTP Authorization" header. Check this box to use the "HTTP Authorization" header. This box should be checked when connecting with another Now SMS/MMS gateway.

"**Server Type**" provides pre-defined templates for connections to common gateway interfaces. If you are connecting to one of the servers with a pre-defined template, such as another Now SMS/MMS gateway, select its server type here. Otherwise, select "Custom" to define a custom template.

"**URL Template Text**" is a URL template that is used when sending text SMS messages. When the gateway has a text SMS message to send, it connects to the HTTP server and issues the URL request specified in this field, replacing the "replaceable parameters" with values for the message to be sent. A complete list of "replaceable parameters" is provided below.

"**URL Template Binary**" is a URL template that is used when sending binary SMS messages. When the gateway has a text SMS message to send, it connects to the HTTP server and issues the URL request specified in this field, replacing the "replaceable parameters" with values for the message to be sent. A complete list of "replaceable parameters" is provided below.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

URL Template Replaceable Parameters:

@@UserName@@	The user name configured for this connection (optional)
@@Password@@	The password configured for this connection (optional)
@@PhoneNumber@@	The phone number of the recipient to receive this SMS message (required)
@@Text@@	The text of the SMS message (required for text messages)
@@Data@@	The data of the SMS message in binary format as a string of hexadecimal characters (either this or @@DataBin@@ required for binary messages)
@@DataBin@@	The data of the SMS message in binary format as the actual binary data in URL escaped format (either this or @@Data@@ required for binary messages)
@@UDH@@	The "User Data Header" of a binary message as a string of hexadecimal characters (either this or @@UDHBin@@ required for binary messages)
@@UDHBin@@	The "User Data Header" of a binary message in binary format as the actual binary data in URL escaped format (either this or @@UDH@@ required for binary messages)
@@PID@@	SMS "Protocol ID" field as a hexadecimal value
@@PIDdecimal@@	SMS "Protocol ID" field as a decimal value
@@DCS@@	SMS "Data Coding Scheme" field as a hexadecimal value
@@DCSdecimal@@	SMS "Data Coding Scheme" field as a decimal value.
@@Sender@@	Phone number to be included as the sender of this message.
@@ServiceType@@	*Advanced:* The "service_type" value associated with the message. This parameter is usually only present if the message was originally submitted by or received from an SMPP client or server, and preserves the original SMPP value.
@@ReceiptRequested@@	*Advanced:* This value is set to Yes if the submitted message requested a return receipt.
@@UDHDestPort@@	*Advanced:* If the UDH of the message includes destination port addressing, this contains the destination port value (in hexadecimal), otherwise blank.
@@UDHDestPortDecimal@@	*Advanced:* If the UDH of the message includes destination port addressing, this contains the destination port value (in decimal), otherwise blank.
@@MessageID@@	*Advanced:* This value contains the NowSMS assigned MessageID, so that it can be passed to the HTTP SMSC as a message reference id.
@@SubmitUser@@	*Advanced:* This value can include the user account that originally submitted the message to NowSMS.
@@SubmitPassword@@	*Advanced:* This value can include the password for the user account that originally submitted the message to NowSMS.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"**Sender Address**" specifies the default sender address (phone number) to apply to outbound SMS messages. The SMSC to which you are connecting may override this setting. (Note: The sender number is only transmitted if the @@Sender@@ variable is included in your URL template string.)

Check "**Allow Sender Address Override**" if you want to allow messages submitted to the gateway to be able to specify a sender address. If this box is checked, and a sender address is present in a message being submitted to an HTTP based SMSC, the sender address in the message will be submitted to the SMSC. The SMSC may override this setting.

Check "**Remove '+' from Recipient Phone Number**" if the gateway should remove the "+" character from international phone numbers before submitting the message to the HTTP SMSC. HTTP SMSC interfaces based upon the Kannel product expect the "+" character to be removed.

Some types of messages processed by the Now SMS/MMS Gateway may require multiple SMS messages to transmit a single logical message. This is because the maximum size of an SMS message is 160 text characters or 140 binary bytes of data. The Now SMS/MMS Gateway automatically segments larger messages and submits them as multiple SMS messages that can be reassembled by the receiving client. Some HTTP based SMSCs prefer to split larger messages themselves. Check "**Send long messages without segmentation**" if you want the SMSC to split larger messages into multiple SMS messages, or leave this setting unchecked to allow the Now SMS/MMS Gateway to perform necessary segmentation of large messages.

By default, when sending a long text message, or a text message that includes UDH (such as an EMS message, or a message that includes source or desintation port addressing), NowSMS will encode the SMS message in a binary format using 7-bit binary encoding for the message text. To use standard text encoding instead, it is necessary to uncheck "**Use 7-bit binary encoding for long text messages**". Unchecking this setting will cause NowSMS to use the "URL Template Text" when sending such a message, so ensure that the template includes a parameter for including the UDH of the message.

By default, when sending a Unicode text message, NowSMS will encode the Unicode text as a hexadecimal string, and pass it to the HTTP SMSC as a binary message. If your SMSC prefers to receive these messages as standard text, it is necessary to uncheck "**Use hex encoding for Unicode messages**". When this setting is unchecked, NowSMS will use the "URL Template Text" when sending a Unicode message, and will encode the text using UTF-8 character set encoding, unless another character set is specified.

"**Character Set**" specifies the character set to be used when transmitting text messages to the HTTP SMSC.

"**Max Connections**" specifies the maximum number of concurrent connections that NowSMS can have open to the HTTP SMSC for transmitting messages. By defining more than one connection, NowSMS will attempt to simultaneously open multiple connections to the HTTP SMSC for sending messages.

The "**Routing**" group of options is used when multiple SMSC connections are defined to the gateway. These options define what messages should be routed to this connection. The Routing options are common to SMPP, GSM Modem, UCP/EMI and HTTP connections, and are described in the **Routing Options** section *(page 52)*.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

When the "**OK**" button is selected, the gateway will attempt to connect to the HTPP server to verify the configuration information provided. Diagnostic information will be displayed if the connection fails. The connection will only be added to the configuration after a successful connection to the HTTP server. Note that although a connection attempt was successful, you should attempt to send a message through the interface to verify that the URL templates are defined correctly.

Receiving SMS Messages via an HTTP SMSC Connection

In most configurations, an HTTP SMSC connection can only be used for sending SMS messages. For both sending and receiving of SMS messages, it is better to use one of the SMS specific protocols, such as SMPP.

However, it is also possible to receive SMS messages using an HTTP SMSC connection, so that they will be routed to a 2-way command processor, or even to a local user account.

To submit a received SMS message into NowSMS, so that the message will be routed to the 2-way command processor, it is necessary to make an HTTP connection to NowSMS in the same way that an application would submit a message to NowSMS for outbound message delivery. For more information on the URL format for this HTTP connection, please see **Submitting SMS Messages - URL Parameters** on page 187.

To indicate that the message is an inbound received message, instead of an outbound message, it is necessary to add "&InboundMessage=Yes" to a standard NowSMS URL request.

Alternatively, it is possible to include "&LocalUser=XXXXX" to route a message to a local "SMS Users" account for delivery via SMPP.

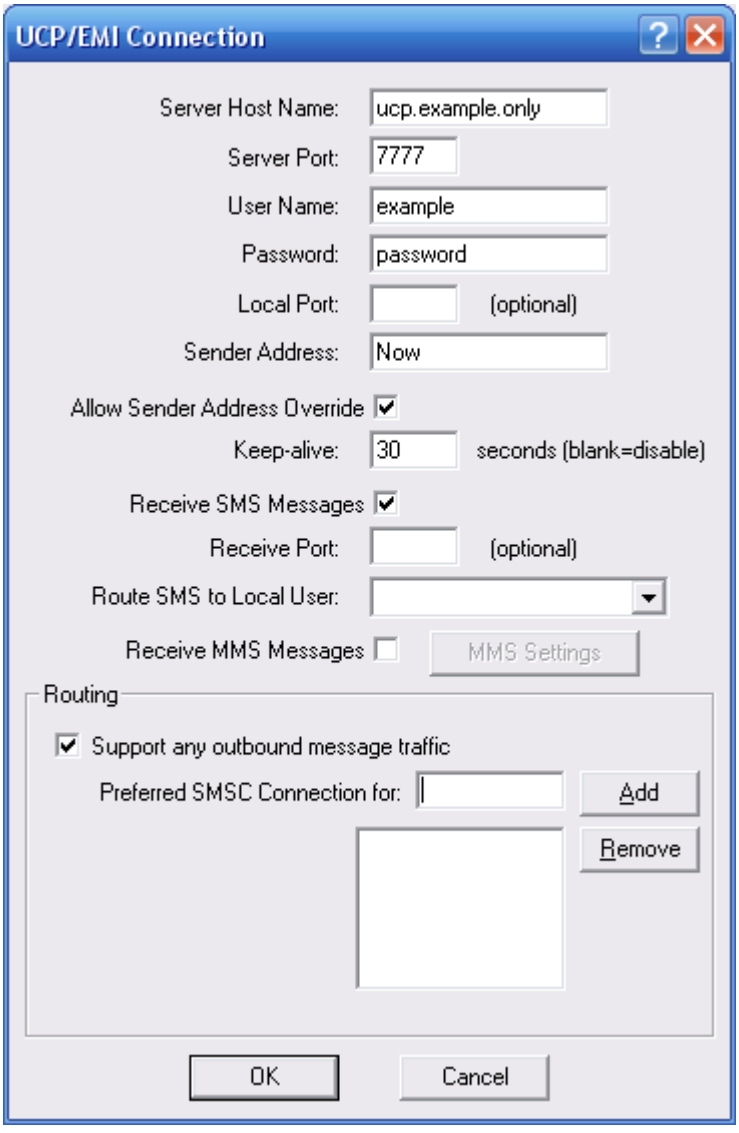
Note that if "**SMS Users**" accounts are enabled in NowSMS, it is necessary to include an "&User=xxxx&Password=yyyy" parameter to authenticate as an "SMS Users" account, even though the messages are to be processed as received messages instead of outbound message submissions. To prevent the possibility of a received message being routed for outbound delivery, we recommend creating a separate "SMS Users" account for this type of message delivery. Check "Enable Credit Balance" for the account, and leave the credit balance at 0. This will allow the account to deliver received SMS messages (i.e., "&InboundMessage=Yes"), but will disallow any attempts to submit a message for outbound delivery.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

UCP/EMI SMSC

The gateway supports the UCP/EMI (Universal Computer Protocol / External Machine Interface) protocol, version 3.5 or higher, to connect to an SMSC over the internet or other private TCP/IP network. Some mobile operators provide UCP connections for higher volumes of SMS traffic.

To add a UCP/EMI connection, select "**Add**" from the "**SMSC**" configuration dialog. Then select "**UCP/EMI over TCP/IP**".



"**Server Host Name**" specifies the TCP/IP address or host name of the UCP/EMI server.

"**Server Port**" specifies the TCP/IP port on the UCP/EMI server to which the gateway should connect.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"**User Name**" specifies the user name for the gateway to use when connecting to the UCP/EMI server.

"**Password**" specifies the password for the gateway to use when connecting to the UCP/EMI server.

Note: Some UCP/EMI systems may not require a username and password, and validate your account based only on the TCP/IP address of your system. If the username and password fields are left blank, the gateway will not send a UCP-60 bind message to the server to login.

"**Local Port**", if set, should be set to the value of a TCP/IP port number on the gateway PC. If set, the gateway will initiate all connections to the UCP/EMI server from this port.

"**Sender Address**" specifies the default sender address (phone number) to apply to outbound SMS messages. The UCP/EMI server may override this setting.

Check "**Allow Sender Address Override**" if you want to allow messages submitted to the gateway to be able to specify a sender address. If this box is checked, and a sender address is present in a message being submitted to an UCP/EMI based SMSC, the sender address in the message will be submitted to the SMSC. The UCP/EMI server may override this setting.

Some UCP/EMI servers may require that clients submit keep-alive messages to the server every so many seconds or minutes, or the server will time out the connection. To enable keep-alive messages, specify a value in seconds for the "**Keep-Alive**" setting. (Note: The gateway uses a UCP-31 message for the keep-alive message.)

Check the "**Enable Receive Messages**" box if you wish to receive messages from the UCP/EMI server. When this box is checked, the gateway can either receive messages using a single connection to the UCP/EMI server, or the "**Receive Port**" setting can specify a TCP/IP port number on the gateway PC that will listen for connections from the UCP/EMI server, and receive messages. (Note: The "Receive Port" should be left blank for most configurations. This setting should only be specified if the UCP/EMI service provider will initiate connections to your server when it has a message to deliver, which is a rare configuration. Most service connections require that your server initiate all connections to the service provider, and the "Receive Port" setting is not used in those configurations.)

It is possible to configure NowSMS to route all SMS messages received via this SMSC connection to a user account on the NowSMS server by selecting the user account in the "**Route SMS to local user**" setting. It is possible for a user to connect to the NowSMS server either using SMPP (an SMS specific protocol) or POP3 (an e-mail protocol). These user accounts are defined on the "**SMS Users**" page of the NowSMS configuration dialog *(see page 64)*.

Check the "**Receive MMS Messages**" box if you will be receiving MMS Notification messages via the SMSC. Note that ***most service providers do not support*** the routing of MMS notification messages via an SMSC connection. If MMS messages will be received via the SMSC connection, it is necessary to also configure additional MMS settings. The "**MMS Settings**" dialog will display a dialog with additional configuration settings that are required for enabling the gateway to be able to receive MMS messages from the operator network. For additional configuration information, please refer to **Connecting to an**

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Operator MMSC – Using a GPRS Modem on page 136. For additional configuration information on processing received MMS messages, please refer to **2-Way MMS Support** on page 240.

The "**Routing**" group of options is used when multiple SMSC connections are defined to the gateway. These options define what messages should be routed to this connection. The Routing options are common to SMPP, GSM Modem, UCP/EMI and HTTP connections, and are described in the **Routing Options** section *(page 52)*.

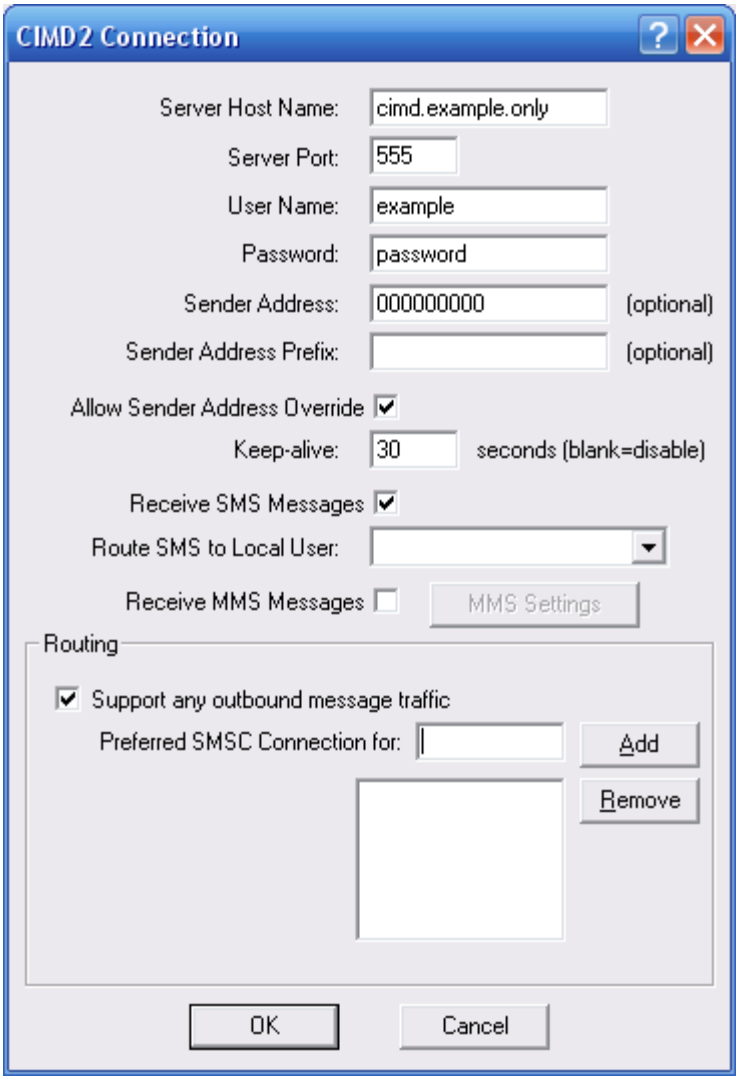
When the "**OK**" button is selected, the gateway will attempt to connect to the UCP/EMI server to verify the configuration information provided. Diagnostic information will be displayed if the connection fails. The connection will only be added to the configuration after a successful connection to the UCP/EMI server.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

CIMD2 SMSC

The gateway supports the CIMD2 (Computer Interface to Machine Distribution, version 2) protocol to connect to an SMSC over the internet or other private TCP/IP network. This protocol is implemented by Nokia SMSCs.

To add a CIMD2 connection, select "**Add**" from the "**SMSC**" configuration dialog. Then select "**CIMD2 over TCP/IP**".



"**Server Host Name**" specifies the TCP/IP address or host name of the CIMD2 server.

"**Server Port**" specifies the TCP/IP port on the CIMD2 server to which the gateway should connect.

"**User Name**" specifies the user name for the gateway to use when connecting to the CIMD2 server.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"**Password**" specifies the password for the gateway to use when connecting to the CIMD2 server.

"**Sender Address**" specifies the default sender address (phone number) to apply to outbound SMS messages. The CIMD2 server may override this setting.

Many CIMD2 connections allocate multiple phone numbers to an individual SMSC account. The "**Sender Address Prefix**" setting specifies the prefix associated with all phone numbers allocated to the SMSC account.

Check "**Allow Sender Address Override**" if you want to allow messages submitted to the gateway to be able to specify a sender address. If this box is checked, and a sender address is present in a message being submitted to a CIMD2 based SMSC, the sender address in the message will be submitted to the SMSC. The CIMD2 server may override this setting.

Some CIMD2 servers may require that clients submit keep-alive messages to the server every so many seconds or minutes, or the server will time out the connection. To enable keep-alive messages, specify a value in seconds for the "**Keep-Alive**" setting.

Check the "**Enable Receive Messages**" box if you wish to receive messages from the CIMD2 server.

It is possible to configure NowSMS to route all SMS messages received via this SMSC connection to a user account on the NowSMS server by selecting the user account in the "**Route SMS to local user**" setting. It is possible for a user to connect to the NowSMS server either using SMPP (an SMS specific protocol) or POP3 (an e-mail protocol). These user accounts are defined on the "**SMS Users**" page of the NowSMS configuration dialog *(see page 64)*.

Check the "**Receive MMS Messages**" box if you will be receiving MMS Notification messages via the SMSC. Note that ***most service providers do not support*** the routing of MMS notification messages via an SMSC connection. If MMS messages will be received via the SMSC connection, it is necessary to also configure additional MMS settings. The "**MMS Settings**" dialog will display a dialog with additional configuration settings that are required for enabling the gateway to be able to receive MMS messages from the operator network. For additional configuration information, please refer to **Connecting to an Operator MMSC – Using a GPRS Modem** on page 136. For additional configuration information on processing received MMS messages, please refer to **2-Way MMS Support** on page 240.

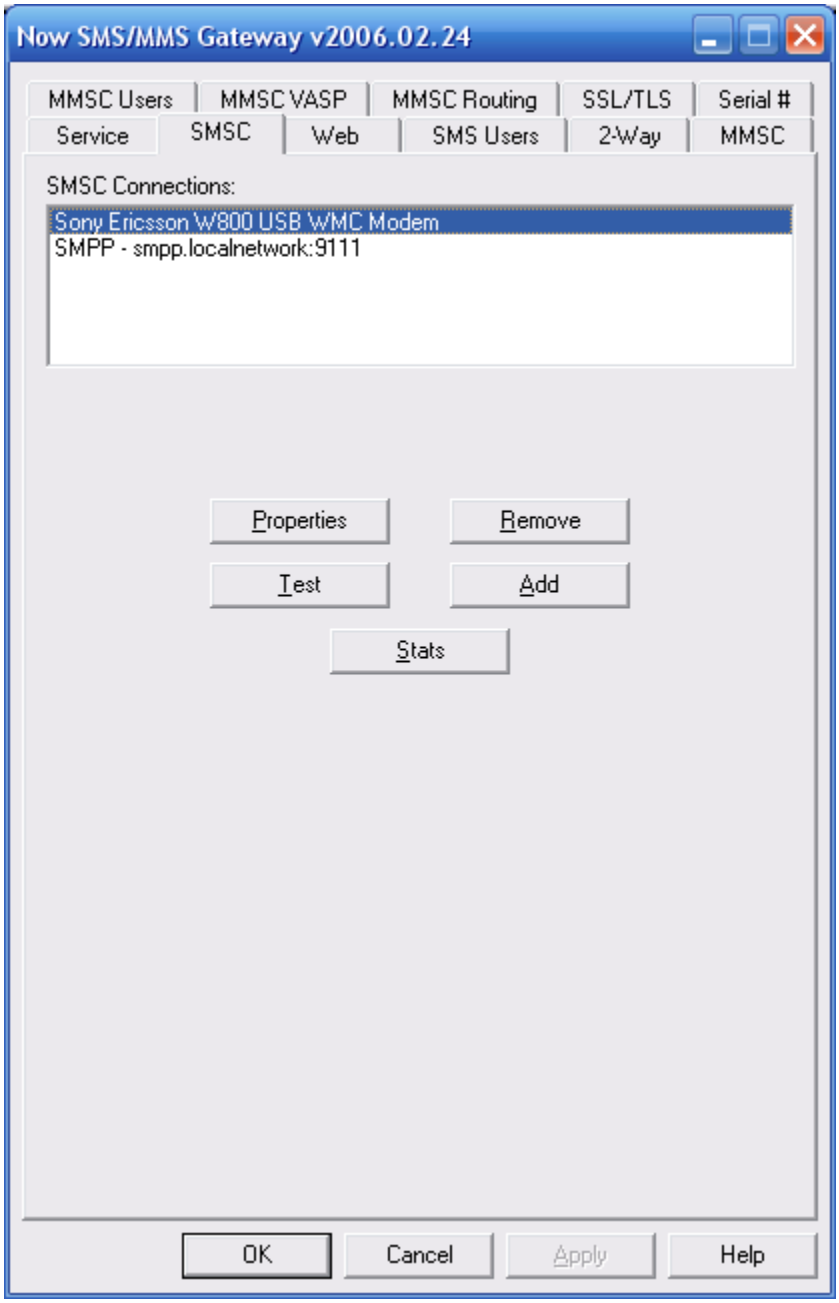
The "**Routing**" group of options is used when multiple SMSC connections are defined to the gateway. These options define what messages should be routed to this connection. The Routing options are common to SMPP, GSM Modem, UCP/EMI and HTTP connections, and are described in the **Routing Options** section *(page 52)*.

When the "**OK**" button is selected, the gateway will attempt to connect to the UCP/EMI server to verify the configuration information provided. Diagnostic information will be displayed if the connection fails. The connection will only be added to the configuration after a successful connection to the UCP/EMI server.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Additional SMSC Configuration Options

Once one or more modems are defined, additional options are displayed under the "**SMSC**" tab.



The "**Properties**" button allows you to configure properties for the selected connection. For SMSC connections other than GSM modems, the standard configuration dialogs appear. For modem connections, the **Routing Options** dialog *(page 52)* will be displayed, defining what SMS message recipients should be routed via this connection.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The "**Test**" button allows you to test the selected connection and confirm that the gateway software is still able to communicate properly with the modem or service. Note that if the gateway service is currently active, the service will be temporarily stopped while the test is running.

The "**Remove**" button allows you to remove the selected connection from the configuration, so that the gateway will no longer attempt to use the connection.

After changes are made, use the "**Apply**" button to save any changes. Use the "**Ok**" button to close the dialog.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

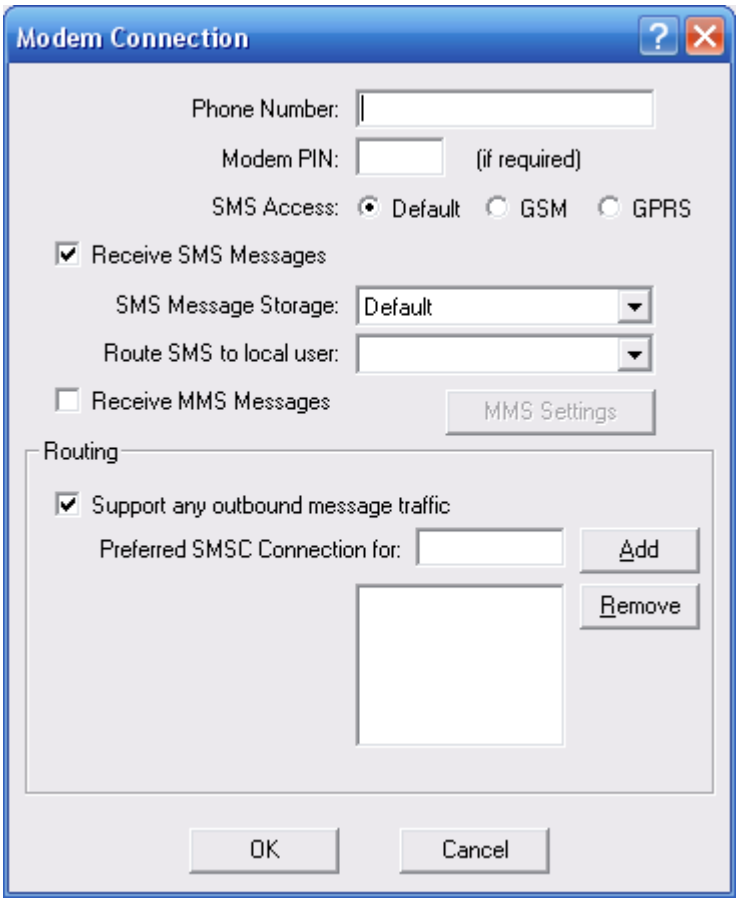
Routing Options

The **"Routing Options"** dialog for a modem connection contains a field to set the **Phone Number** of the GSM modem for that connection. This field is not required, but if the field is set, then the following will be enabled:

- Inbound messages arriving on that GSM modem connection will be associated with this phone number. For 2-way SMS applications, the phone number is included in the @@Recip@@ parameter. Additionally, all inbound messages arriving on this interface can be queued for delivery to an SMPP Client account that uses this gateway as its SMPP Server.

- When a sender address is included in a message submitted to the gateway for delivery, and the sender address matches the configured phone number for a particular GSM modem connection, the gateway will ensure that the message is sent via this GSM modem.

"**Routing Options**" are available for all SMSC connections defined to the gateway. For SMPP, UCP/EMI and HTTP connections, the "**Routing**" group of options is displayed on the standard configuration dialog. For modem connections, a separate dialog is displayed when "**Properties**" is selected for the connection.



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If "**Support any outbound message traffic**" is checked, this connection is available to route any messages, unless the recipient of the message appears as a "**Preferred SMSC Connection**" for another connection.

If "**Support any outbound message traffic**" is not checked, this connection will only route messages when the recipient of the message appears in the "**Preferred SMSC Connection**" list for this connection.

The "**Preferred SMSC Connection**" list specifies one or more patterns to match to determine if a message should be routed by a particular connection. Patterns consist of a phone number string, and can include the wildcard characters "*" and "?". The "*" character matches any number of characters, and the "?" character matches any single character. When a pattern is defined on a preferred connection list, it means that any messages to recipients that match this pattern will be routed ONLY by this connection (unless another connection shares the same preferred connection pattern).

In the example above, the connection will route any messages to recipients in the "+44" country code (because "+44*" is on the preferred SMSC connection list for this connection). Also, this connection will route messages for any other recipients where the recipient does NOT match a pattern on the preferred list for any other connection.

In the example above, any other defined connections would not route messages for recipients in the "+44" country code, unless "+44*" was repeated on the preferred connection list for another connection.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

SMS Message Routing Logic

For the sake of clarity, this section will define the routing logic that NowSMS uses to determine which connection should be used for routing an SMS message for outbound delivery.

When NowSMS routes a message, it first looks to see if a sender address has been specified for the message submission (normally there is not a sender address specified, unless you submitted the message via HTTP and specified a "Sender=" parameter). If a sender address was specified, then NowSMS checks to see if the sender address matches the "**Default Sender Address**" that is configured for any of the SMSC links (or the "**Phone Number**" associated with a GSM modem). If NowSMS finds a match, then it will route the message only via an SMSC connection with a matching sender address.

If NowSMS did not find a match on the sender address, then it evaluates the recipient address, and it will look to see if it finds a match in the "**Preferred SMSC Connection for**" recipient address masks associated with any of the SMSC connections. (These recipient address masks can be wildcards such as "+44*" to match any phone number that starts with "+44".) If NowSMS finds a match, then it looks for the longest mask that provides a match, and routes the message via the connection with the longest matching mask. (For example, if you were sending to +441624999999, and you had one connection with a mask of "+44*", and another with "+441624*", then the connection with the mask of "+441624*" would be used as it is a longer match than "+44*".)

Patterns consist of a phone number string, and can include the wildcard characters "*" and "?". The "*" character matches any number of characters, and the "?" character matches any single character.

If there is no match on the recipient address mask, then the message will be routed via any connection that has "Support any outbound message traffic" checked.

It should also be mentioned that if NowSMS found multiple matches on the sender address, it evaluates the "Preferred SMSC Connection for" recipient address masks for each of the connections that had a sender address match.

Beginning with the NowSMS 2007 release, an HTTP parameter setting has been added that can be used to explicitly route a message via a particular SMSC, so that it is not necessary to always use the sender/recipient matching logic, if it is not appropriate for your configuration.

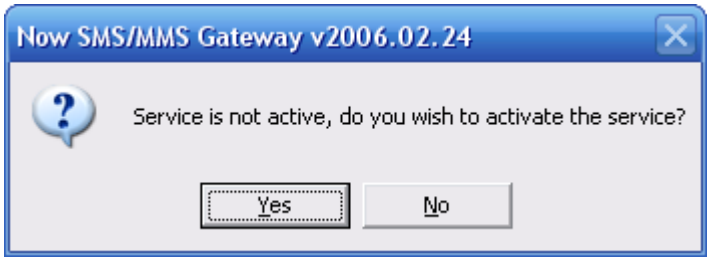
When a message is submitted via HTTP, the HTTP interface supports a parameter of "&SMSCRoute=xxxxx", where the value of this setting can be the name of a defined SMSC (e.g., "Bluetooth Modem" or "SMPP - a.b.c.d:xyz"). Or, rather than using the SMSC name, it can be a route name that is defined as associated with one or more SMSCs. To define a route name for an SMSC, it is necessary to manually edit SMSGW.INI, and under the appropriate section header (e.g., [Modem - Bluetooth Modem] or [SMPP - a.b.c.d:xyz]), add RouteName=xxxxx. It is possible for multiple SMSCs to share the same route name, meaning that if a message is submitted via HTTP with the "&SMSCRoute=xxxxx" parameter, it will be routed outbound over the first available SMSC that is configured with the RouteName=xxxxx setting.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Running as a Service

In the Windows environment, service processes are started automatically when the PC is started, so that it is not necessary for a user to logon to the computer to run a service program. The Now SMS/MMS gateway operates as a Windows service.

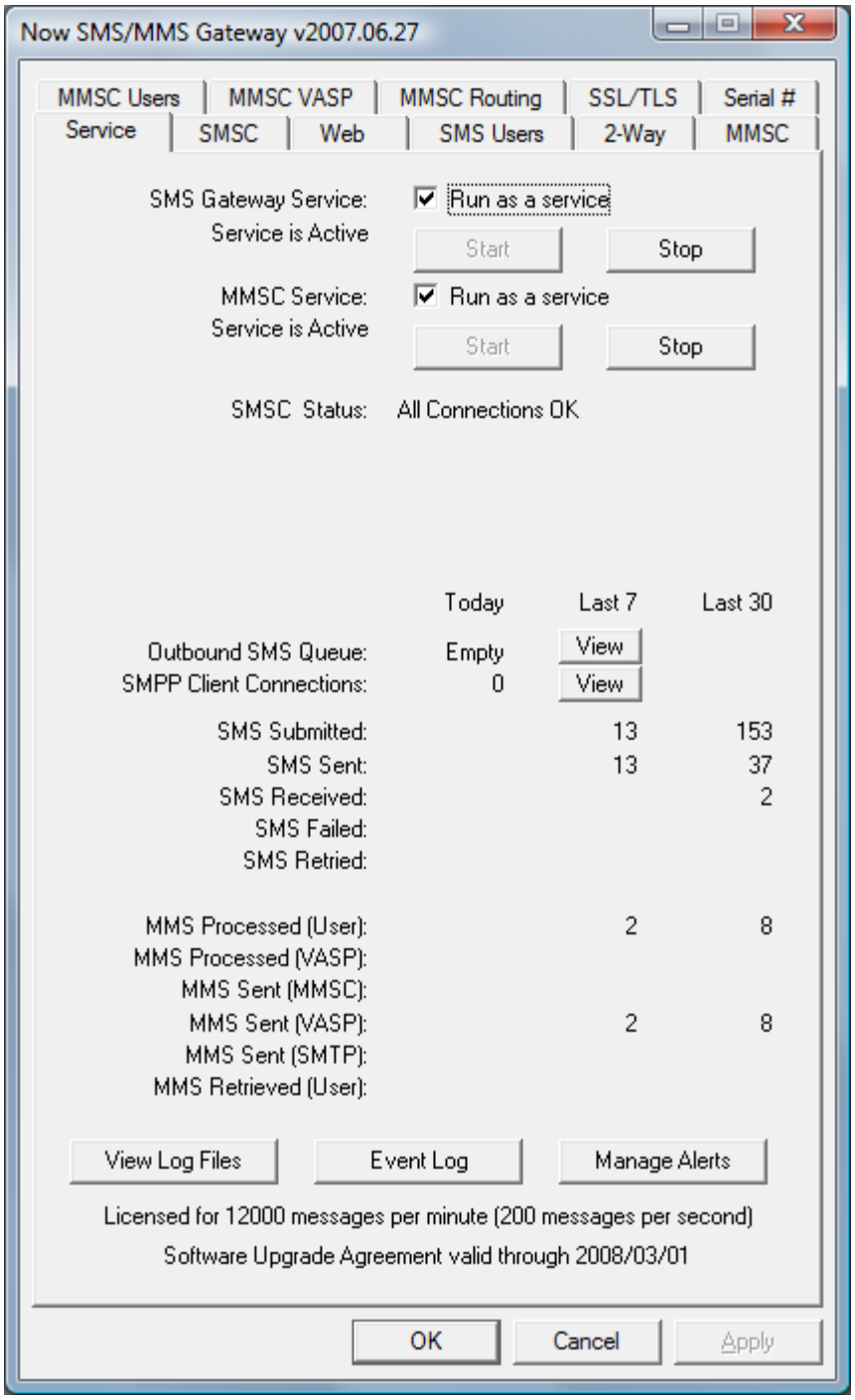
When you are configuring the gateway, and select "**Ok**" to close the configuration dialog, the gateway checks to see if the service process is active. If the service process is not active, the following dialog is displayed:



Select "**Yes**" to install and activate the service, or "**No**" to exit without activating the service.

The "**Service**" configuration dialog can also be utilized to install or remove the gateway services.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



NowSMS installs as two separate services, for historical reasons. Most installations will want to have both services installed, even if they are only making use of only SMS or MMS related functionality.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The "**SMS Gateway Service**" provides functionality for supporting the following features:

- Web Menu Interface
- Submission of SMS Messages and SMSC Connectivity
- Now SMS/MMS Proprietary URL Submission of MMS Messages
- SMPP Server
- 2-Way SMS

The "**MMSC Service**" provides functionality for supporting the following features:

- MMSC: MM7, MM4, MM1 and EAIF protocol handling
- MM1 protocol handling for GPRS modem connections
- SMTP inbound and outbound messaging (including alerting for system events)
- POP3 Server
- E-Mail to SMS
- Multimedia WAP Push

In order to function properly, the "SMS Gateway Service" requires that an available unique TCP/IP port number be assigned to the NowSMS web interface. This port number is defined in the "Port number for web interface" setting on the "Web" configuration dialog. For more information, see **Configuring the Web Interface** on page 60.

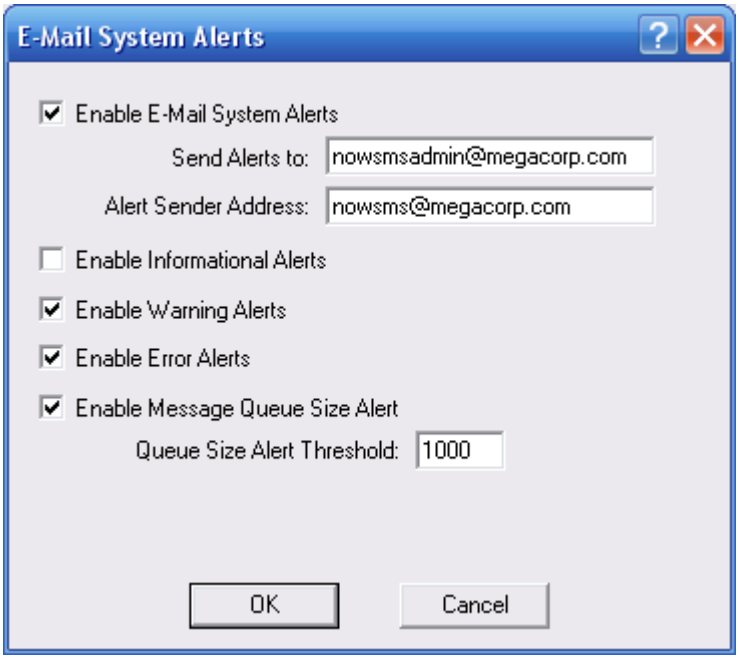
In order to function properly, the "MMSC Service" requires that an available unique TCP/IP port number be assigned to the HTTP interface of the MMSC. This port number is defined in the "HTTP Port Number" setting on the "MMSC" configuration dialog. For more information, see **MMSC Messaging Server** on page 122.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Event Log and System Alerts

The "**Event Log**" button displays a dialog that shows events related to the gateway which have been logged to the Windows Event Log. The gateway always logs when its services are started and stopped. Additionally errors are reported to the event log if a particular SMSC interface fails to initialize, and when a user account exceeds their defined message quota for sending messages.

These events can also be sent by NowSMS to one or more e-mail addresses to alert system administrators of potential problems. To configure these alerts, press the "**Manage Alerts**" button.



As the alerting function uses e-mail, and all e-mail functionality within NowSMS is provided by the MMSC Service, this functionality requires that the MMSC Service be activated.

Check the "**Enable E-Mail System Alerts**" button to enable e-mail alert functionality.

The "**Send Alerts to**" field can contain a comma-delimited list of one or more recipients to receive the e-mail alerts

The "**Alert Sender Address**" field specifies the address that will appear as the "From:" address in alert messages.

Check "**Enable Informational Alerts**" to enable alerts for informational events generated by the NowSMS gateway. Examples of informational alerts include start-up and shut-down messages when the NowSMS services are started or stopped, as well as when an SMSC interface is successfully re-initialised after an error condition has occurred.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Check "**Enable Warning Alerts**" to enable alerts for warning events generated by the NowSMS gateway. Examples of warning alerts include user accounts exceeding their allocated messaging quotas, as well as SMS messages being rejected by the upstream SMSC connection.

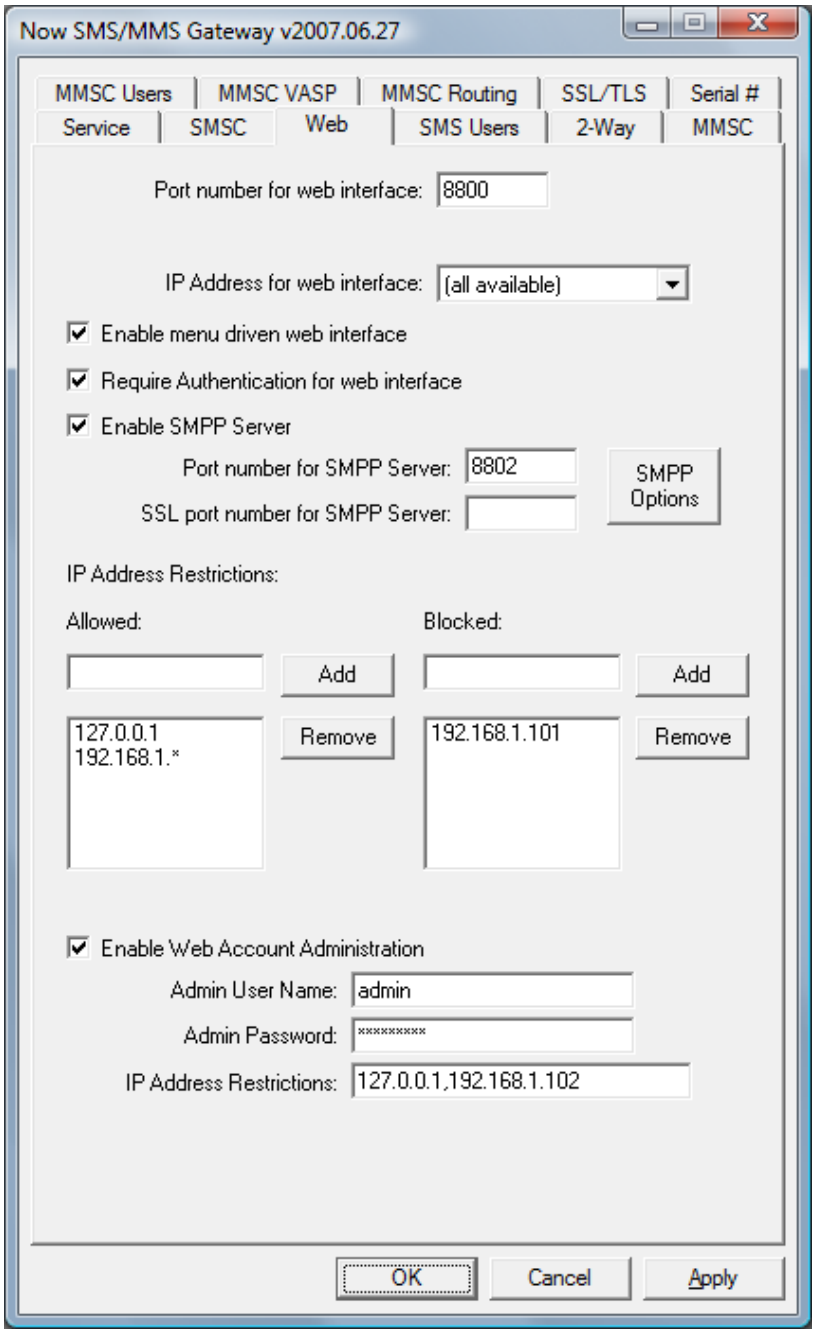
Check "**Enable Error Alerts**" to enable alerts for error events generated by the NowSMS gateway. Examples of error events include failure to connect to a configured SMSC, failure to start a service due to TCP/IP port conflicts.

Check "**Enable Message Queue Size Alert**" to define a "**Queue Size Alert Threshold**", where an alert will be sent if the number of messages in the outbound SMS messaging queue exceeds the defined threshold.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Configuring the Web Interface and SMPP Server

When you wish to submit any type of SMS message, including MMS notifications or WAP push messages, you must submit the request via the gateway's web interface. The "**Web**" configuration dialog contains settings relevant to this web interface.



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"**Port number for web interface**" - Configure the gateway to listen for web/HTTP requests on a particular TCP/IP port number. The standard web port on the internet is 80, however you will most likely want to use a non-standard port for this service. The service defaults to the non-standard port 8800.

The PC that is running the gateway might have other web services installed. For this reason, the gateway allows you to specify which of the available IP addresses on the current PC should be used by the gateway. The "**IP Address for web interface**" prompt displays the available IP addresses on the current PC. To make the gateway service available via any address on the current PC, select "(all available)", otherwise select a specific IP address.

"**Enable menu driven web interface**" specifies whether or not an HTML menu will be displayed when a user connects to the gateway via the web interface. This menu interface can be helpful when you are first exploring the features of the gateway. If this menu setting is disabled, then the gateway will require the appropriate URL parameters to perform any tasks. These URL parameters are defined elsewhere in this document in the section titled **Submitting SMS Messages - URL Parameters** on page 187.

Checking "**Enable SMPP Server**" enables the SMPP Server module of the gateway. While the gateway has the ability to act as an SMPP client, sending and receiving messages through an external SMPP server, it also has the ability to act as an SMPP server to provide message sending and receiving services to other SMPP clients. When you enable the SMPP server, you must specify a TCP/IP "**Port number for SMPP Server**". This is a local port number on the gateway PC, which must not be in use by any other applications. The gateway will listen for SMPP clients to connect to the gateway on the port specified. User accounts for SMPP clients are defined on the "**SMS Users**" configuration dialog *(page 64)*.

The "**SMPP Options**" button displays some configuration options for the SMPP Server, and is described at the end of this section.

To prevent unauthorized access to the gateway, access should be limited. For most applications, it is recommended that this gateway be installed inside of your firewall, to help ensure that users on external computers cannot access the gateway. As further authorization measures, the gateway can also limit access by restricting address to a limited range of IP addresses, and by requiring a username and password for access.

To define that a username and password be required for access to the gateway, check "**Require Authentication for web interface**". HTTP requests must include this username and password in order to issue requests to the gateway. (Note: A standard web browser will prompt for the username and password.) The "**SMS Users**" configuration dialog defines user accounts that can access the gateway *(page 64)*.

To restrict access to the gateway to a limited range of IP addresses, "Allowed" and "Blocked" lists may be defined. When the gateway receives a new web request, it consults the "Allowed" and "Blocked" lists to determine if web access is allowed from the IP address of the machine that issued the request.

If an address is listed on the "Blocked" list, access will be denied, and the web interface cannot be used to submit an SMS message from that address.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If an address is not listed on the "Blocked" list, and an "Allowed" list is not defined, the web interface can be used to submit an SMS message from that address.

If an address is not listed on the "Blocked" list, and an "Allowed" list is defined, access will be denied if the address is not included in the "Allowed" list.

To add addresses to either list, enter an address in the appropriate text edit area and press the "Add" button. To remove an address from either list, highlight the appropriate address in the list, and press the "Remove" button. Wildcards can be used in a defined IP address to specify that any value in that portion of the IP address will be considered a match. (For example, 192.168.1.* would be considered a match with both 192.168.1.1 and 192.168.1.100.)

It is also possible to define IP address restrictions for individual "SMS Users" accounts.

Checking **"Enable Web Account Administration"** enables a web-based administration interface that allows for adding, modifying and deleting user accounts that are defined to the NowSMS server. Both "SMS Users" and "MMSC Users" can be edited via this web interface.

To enable this interface, you must define both an **"Admin User Name"** and **"Admin Password"**. This user name and password must be supplied upon attempts to enter the admin interface.

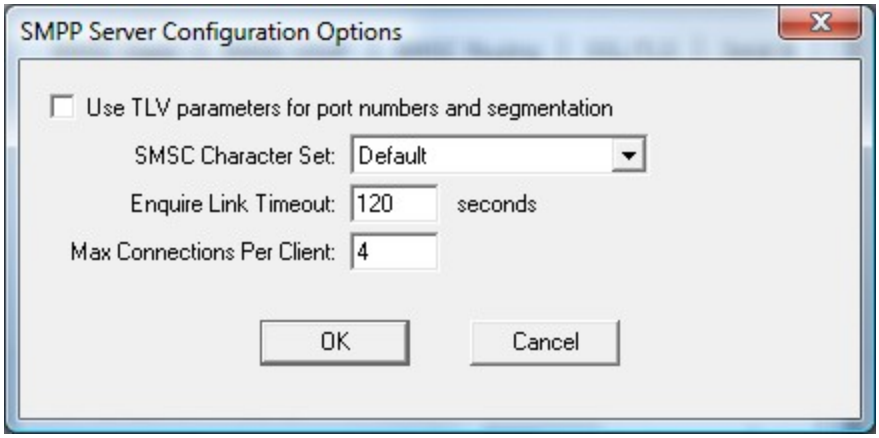
It is also possible to restrict access to the web-based administration interface, so that it can only be access from a list of well-known IP addresses, by entering a comma delimited list of IP addresses in the **"IP Address Restrictions"** field. These IP Address Restrictions apply only to the web-based administration interface. Wildcards can be used in a defined IP address to specify that any value in that portion of the IP address will be considered a match. (For example, 192.168.1.* would be considered a match with both 192.168.1.1 and 192.168.1.100.)

When the Web Account Administration interface is enabled, it is accessible by connecting to the "Port number for web interface" with a relative URI of "/ADMIN". For example, http://127.0.0.1:8800/ADMIN.

The Web Account Administration interface allows for the creating, editing and deleting of "SMS Users" and "MMSC Users" accounts. It also displays statistical information about the server, and provides remote access to NowSMS log files.
The **"SMPP Options"** button displays some configuration options for the SMPP Server.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

SMPP Server Options



When delivering long text messages that must be segmented over multiple SMS messages, or when delivering messages that include port numbers such as for WAP push, NowSMS encodes the message with the appropriate GSM User Data Header (UDH) fields. When the **"Use TLV parameters for port numbers and segmentation"** option is enabled, NowSMS will instead use optional TLV parameters in the SMPP header for port numbers and segmentation. Specifically port numbers will be encoded in the source_port and destination_port parameters, and segmentation will be encoded in the sar_msg_ref_num, sar_total_segments, and sar_segment_seqnum parameters. Regardless of the setting of this parameter, SMPP clients can either encode messages for submission to NowSMS using either GSM UDH or TLV parameters. This setting only controls which format NowSMS uses when delivering messages back to SMPP clients.

SMSC Character Set – By default, NowSMS uses the GSM character set when delivering SMS messages via SMPP, and it indicates via the data_coding parameter that the default encoding is used. If you experience a problem where @ symbols and other characters do not appear correctly in received messages, your SMSC client might be expecting a different character set. Assuming that you are working in a GSM environment primarily, first try changing this setting to "IA5 (GSM)". When this setting is applied, NowSMS will still use the GSM character set, but it will set a flag in the header to indicate this. If the character problems persist, change this setting to "iso-8859-1 (Latin)", which is the standard character set used in Western Europe.

Enquire Link Timeout – SMPP clients are expected to periodically send enquire_link commands when connected to an SMSC. These commands are used to tell the SMSC that the client is still alive and functioning properly. If NowSMS does not receive an enquire_link command, or some other activity, from a connected SMPP client within this timeout period, NowSMS will automatically disconnect and release the client connection. It is not recommended to set a value for this parameter less than 60 seconds. However, a special setting of 0 can be used to disable this timeout, allowing clients to remain connected without sending the enquire_link command.

Max Connections Per Client – This specifies the maximum number of simultaneous connections allowed from any one SMPP client account (although it is possible to override this value on a per account basis). Note that many SMPP clients create two separate connections, 1 for sending and 1 for receiving, so a value of 1 may cause problems.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Defining SMS User Accounts

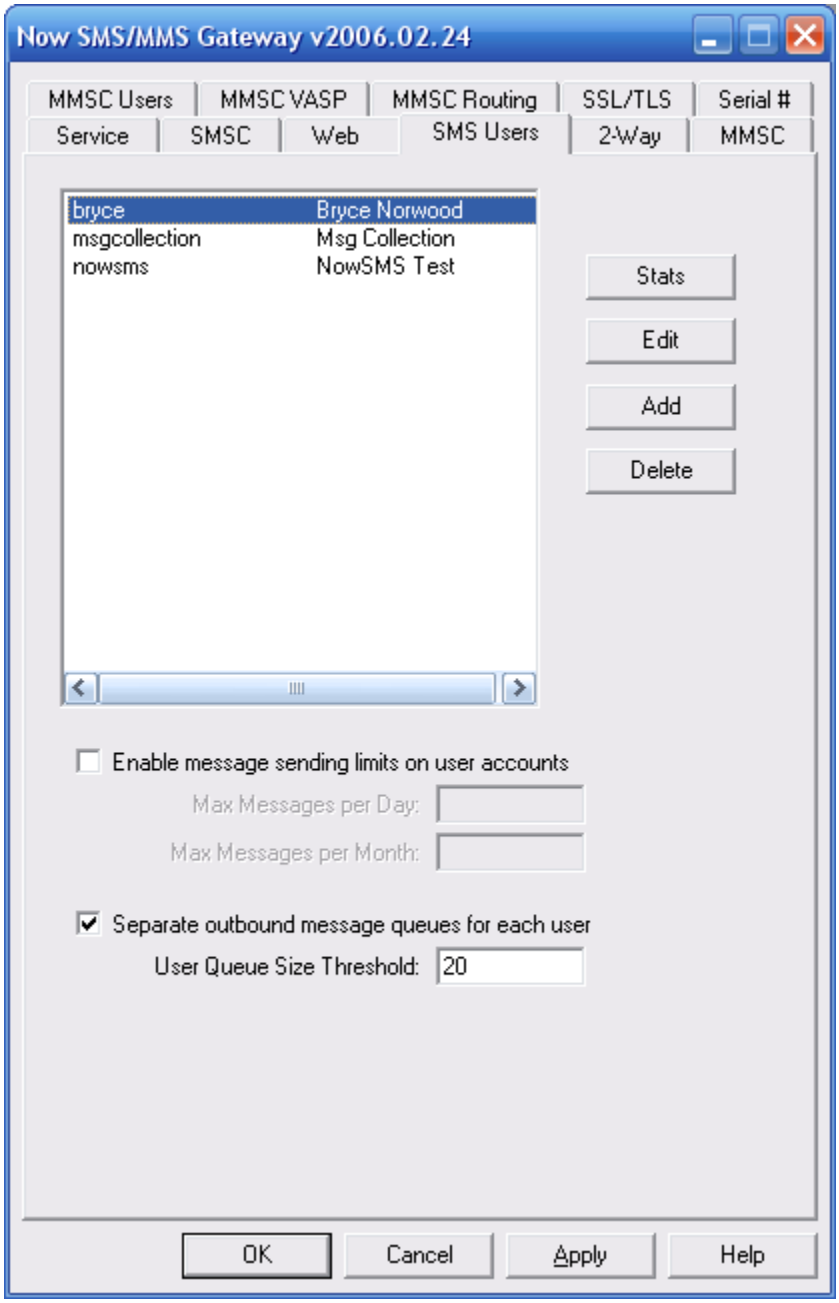
The **"SMS Users"** configuration dialog defines user accounts that are allowed to submit SMS and MMS messages through the gateway. Messages can be submitted to the gateway via HTTP (web interface), SMPP and/or SMTP (e-mail).

(Note: Mobile phone users that connect to the MMSC to send and receive MMS messages are configured under the "MMSC Users" dialog tab. For more information, see page 122.)

When SMS user accounts are defined to the gateway, it is possible to define limits on the number of messages that an account is allowed to submit per day, and per month. It is also possible to define how the account is allowed to connect to the gateway, that is to say whether the account can login via HTTP, SMPP and/or SMTP.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The "**SMS Users**" configuration dialog lists defined user accounts.



If you want to enable sending limits on your user accounts, you must check "**Enable message sending limits on user accounts**", and specify a default limit for the maximum number of messages per day and per month, that users will be allowed to submit.

Individual user accounts can be allowed to have limits higher or lower than the default settings. However, you must enable the sending limits, and define default limits, in order to be able to define sending limits on any of the accounts defined to the gateway. After enabling message sending limits, click "**Apply**" to save the change before adding or editing any user accounts.

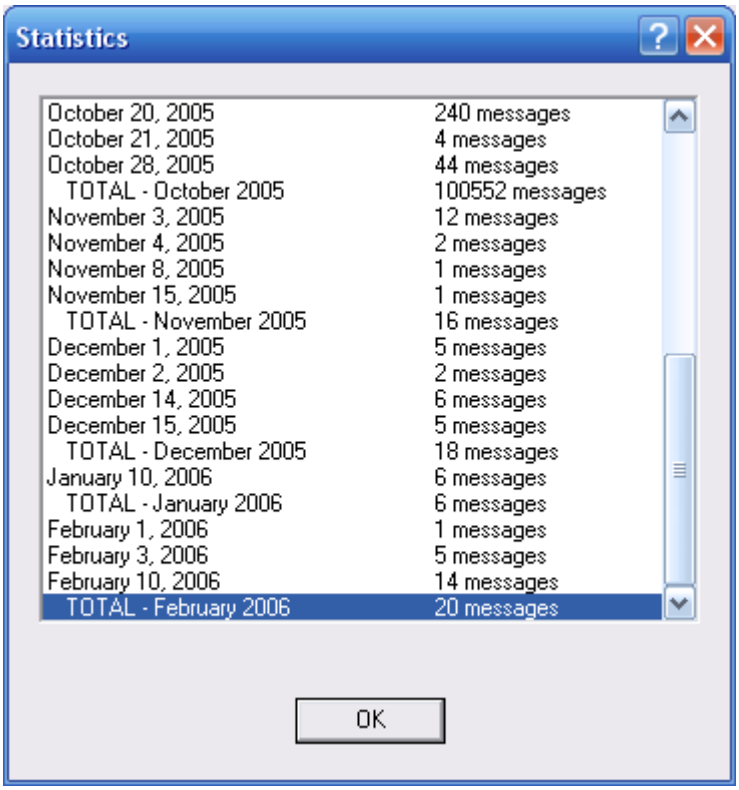
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

By default, NowSMS maintains a single queue for all outbound messages. This means that if "User A" submits 5,000 messages, and "User B" submits a single urgent message, all 5,001 messages are grouped into a single outbound message queue which is processed roughly in the order in which the messages were submitted to the gateway. This may present an unacceptable performance limitation for some multi-user configurations.

To overcome this limitation, it is possible to configure NowSMS to maintain separate outbound message queues for each user account defined to NowSMS. *(When NowSMS is used as a direct-delivery MMSC, this separate outbound message queue setting applies to both "SMS Users" accounts, and "MMSC VASP" accounts.)*

Check "**Separate outbound message queues for each user**" to enable this functionality. When this functionality is enabled, you must define a "**User Queue Size Threshold**" which specifies how many messages will be processed from any one user's outbound message queue before NowSMS switches to another user's queue. It is recommended that this setting not be set too low, as there is system overhead involved in switching between user queues which can slow down overall messaging throughput on systems with faster SMSC connections. 10 is an acceptable minimum value for systems that are using GSM modem connections, while a value of 30 or higher is recommended for systems with faster NowSMS licenses, and correspondingly faster SMSC connections.

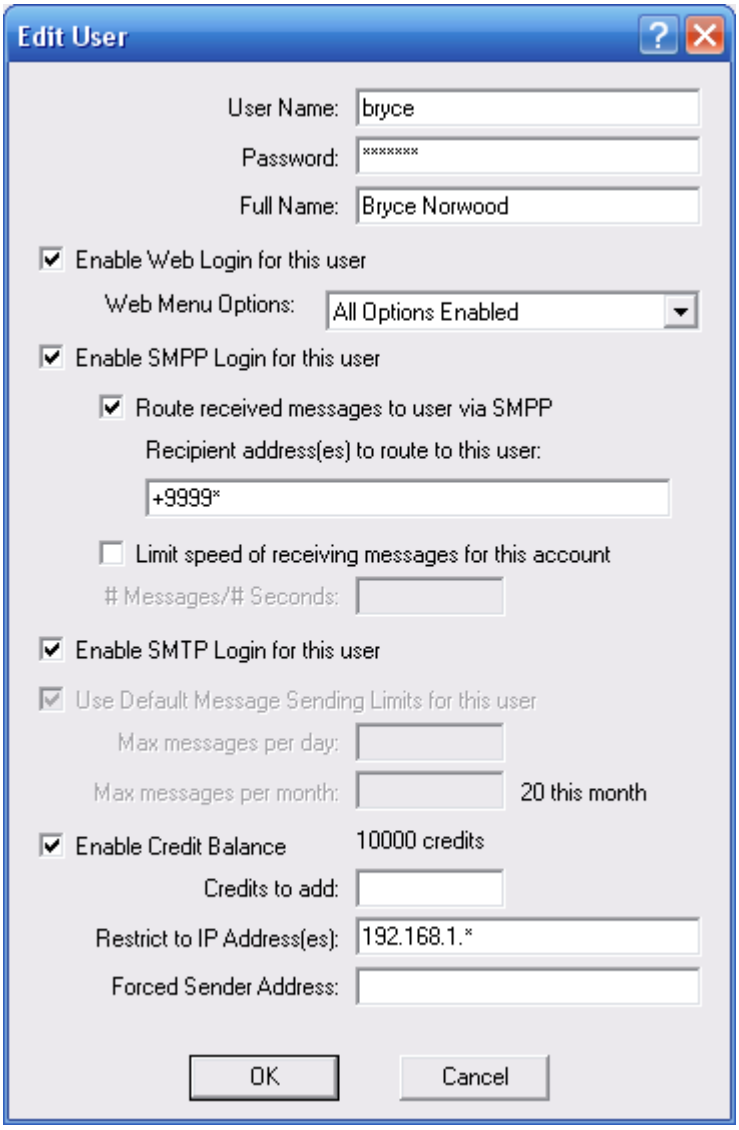
The "**Stats**" button displays information about the number of messages sent by the account that is selected in the list.



The "**Edit**" button is used to edit the settings for a defined user account. The "**Add**" button is used to define a new user account. The "**Delete**" button is used to delete a user account.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

When adding or editing a user account, the following dialog will be displayed:



"**User Name**" and "**Password**" specify the user name and password that will be used to login to the account before sending any messages.

"**Full Name**" specifies a descriptive name for the account.

If the user account should be allowed to log into the web (HTTP) interface to submit messages, check "**Enable Web Login for this user**".

It is also possible to limit which options are displayed on the web user interface on a per-user account basis. Selections include "All Options Available", "Text SMS Only", "SMS, MMS Only", "SMS, WAP Multimedia Only", "SMS, MMS, WAP MM Only". For more information on the web interface, and a better understanding of the functionality available via that interface, refer to the **Web Menu Interface** on page 73.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If the user account should be allowed to connect as an SMPP client to the gateway's SMPP server, check "**Enable SMPP Login for this user**". In addition to allowing the SMPP client to send messages through the gateway, the gateway can also route received messages back to the SMPP client. To enable received messages to be routed to the SMPP client, check "**Route received messages to user via SMPP**", and specify one or more phone numbers (separate multiple phone numbers with a comma), where if the recipient of a message received by the gateway matches one of these phone numbers, the message will be queued for delivery to this SMPP client.

It is possible to limit the speed at which the account is allowed to submit messages to the NowSMS server via SMPP by checking "**Limit speed of receiving messages for this account**". When this option is checked, it is possible to define the number of messages that the account is allowed to submit per second in the "**# Messages/# Seconds**" field. For example, 3 messages per second can be expressed as either "3" or "3/1". 5 messages every 2 seconds can be expressed as "5/2". It is also possible to use this configuration setting to limit the number of simultaneous SMPP connections allowed from this user account. The default value for the maximum number of simultaneous SMPP connections allowed from a single user account is defined under the "**SMPP Options**" setting on the "**Web**" page of the NowSMS configuration dialog (page 60). To override this connection limit for a specific user account, it is necessary to define a limit for the speed at which messages will be received from the account. In the "**# Messages/# Seconds**" field, enter a value of xxx/yyy/zzz, where "xxx" is the number of messages allowed per "yyy" seconds, and "zzz" is the connection limit to be applied for this account. To disable message speed limits, but define a connection limit, use a value of 0/0/zzz, which indicates no limits, but specifies "zzz" as the connection limit for this account.

If the user account should be allowed to log into the SMTP interface to submit messages using an e-mail client, or using POP3 to receive SMS or MMS messages using an e-mail client, check "**Enable SMTP Login for this user**". This allows a user account to login via SMTP with an e-mail client to submit bulk delivery of SMS or MMS messages. The gateway uses the SMTP server from its built-in MMSC to provide this functionality. Therefore the MMSC must be configured and activated to enable this capability. When sending messages in this fashion, the sender must configure an e-mail client to connect to the gateway as an SMTP server, and to use SMTP authentication to login with the defined user name and password. The gateway does not provide an e-mail inbox, only outbound message sending via an SMTP interface. An authenticated SMTP user can send an MMS message by addressing the message to "phonenumber@mms.domain.name", where "mms.domain.name" is the "Domain Name for MMS E-Mail" defined on the MMSC configuration dialog *(page 122)*. An authenticated SMTP user can send an SMS message by addressing the message to "phonenumber@sms.domain.name", where "sms.domain.name" is the "Domain Name for SMS E-Mail" defined on the MMS configuration dialog *(page 122)*.

For more information on e-mail connectivity, please refer to **E-Mail to SMS/MMS Connectivity** on page 243.

If message sending limits are enabled, check "**Use Default Message Sending Limits for this User**" to use the default limits defined for the system, or uncheck this setting to specify a maximum number of messages per day and per month that are allowed to be sent by this account.

It is also possible to define a "**Credit Balance**" for each account. This balance specifies a fixed number of messages that the account is allowed to send. Each time the account

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

sends a message to a recipient, a credit is deducted from this balance. To add or remove credits, enter the number of credits in the "**Credits to add**" field and press "**Ok**". (Prefix the number with a minus symbol, -, to remove credits from an account.)

If the account should only be allowed to connect from a limited set of IP addresses, a comma delimited list of IP addresses from which the account is allowed to login can be entered in the "**Restrict to IP Address(es)**" field. Addresses can include a "*" character as a wildcard to allow connections from all addresses within a subnet.

If all messages submitted by this account should have a particular sender address associated with the message, this sender address can be automatically applied to all messages submitted by the account by specifying a sender address (usually a phone number, but sometimes alphanumeric values are supported by SMSCs) in the "**Forced Sender Address**" field.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

NowSMS in High Availability/Load Balanced Environments

For many configurations it is desirable to install NowSMS on multiple servers in order to achieve fault tolerance and/or improved throughput and performance. NowSMS offers extreme configuration flexibility to facilitate these requirements.

This document outlines the multi-server configuration options that exist for NowSMS v2008.06.03 and later versions.

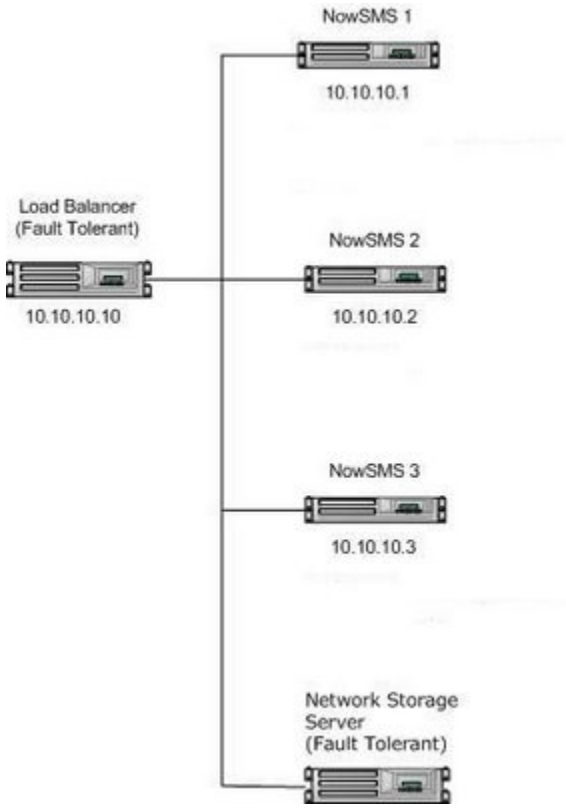
Preferred Solution for Load-Balanced Multi-Server NowSMS Configuration

For most load-balanced multi-server installations it is desirable for each NowSMS server to share the exact same configuration information (SMSC and MMSC connections, user accounts) and message queues.

To run NowSMS in this configuration, the NowSMS program files are installed locally on each server. A load balancer is used to route network traffic to any of the NowSMS servers.

The shared configuration information and message queues are stored on a shared (often fault-tolerant) network storage server.

After installing NowSMS on each of the load balanced servers it is necessary to then create a special file named **SHAREDVOLUME.INI** in the NowSMS program directory. In this file, under a header of [SharedVolume], the following settings are supported:



SharedVolume=\\server\path\

The "SharedVolume" setting specifies a shared directory location under which NowSMS should look for and store all configuration files and message queues.

MessageIDPrefix=xxx

The "MessageIDPrefix" setting specifies a prefix that should be added to all SMS message IDs generated by NowSMS. By specifying a unique value for this setting on each NowSMS server, this ensures that the message IDs that NowSMS generates are unique across each server in a multi-server installation. (Note: For installations that do not use SHAREDVOLUME.INI, it is also possible to specify this setting under the

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

[SMSGW] header of SMSGW.INI.)

LogDirectory=d:\path\

The "LogDirectory" setting specifies a directory under which NowSMS should generate all log files other than debug logs. If not specified, this defaults to the NowSMS program directory. (Note: For installations that do not use SHAREDVOLUME.INI, it is also possible to specify this setting under the [SMSGW] header of SMSGW.INI.)

DebugLogDirectory=d:\path\

The "DebugLogDirectory" setting specifies a directory under which NowSMS should generate any debug log files. If not specified, this defaults to the NowSMS program directory. (Note: For installations that do not use SHAREDVOLUME.INI, it is also possible to specify this setting under the [SMSGW] header of SMSGW.INI.)

Advanced and Custom Multi-Server Configurations

The SHAREDVOLUME.INI solution is ideal for most load-balanced multi-server NowSMS installations. However, for some installations it may be desirable to have slightly different configurations on each server, while sharing some message queues. In those installations, rather than using the SHAREDVOLUME.INI file, settings for individual shared queues or configuration information can be applied in the SMSGW.INI or MMSC.INI file by advanced users.

The following advanced settings are supported for this purpose:

QDir=d:\path or QDir=\\server\path

This setting can be applied in the [SMSGW] section of the SMSGW.INI file to specify the location of the outbound SMS message queue. By default this is the "Q" subdirectory of the NowSMS installation.

BulkQDir=d:\path or BulkQDir=\\server\path

This setting can be applied in the [SMSGW] section of the SMSGW.INI file to specify the location of the outbound bulk SMS message queue. This message queue is used only when the web interface is used to submit the same message to a large number of recipients. By default this is the "BulkQ" subdirectory of the NowSMS installation.

MessageIDTrackingDir=d:\path or MessageIDTrackingDir=\\server\path

This setting can be applied in the [SMSGW] section of the SMSGW.INI file to specify the location of the SMPP receipt message id tracking database. By default this is the "SMPPData" subdirectory of the NowSMS installation.

UsersDir=d:\path or UsersDir=\\server\path

This setting can be applied in the [SMSGW] section of the SMSGW.INI file to specify the location of the "SMS Users" database, which contains all "SMS Users" account information and pending message queues. By default this is the "Users" subdirectory of the NowSMS installation and the SMSUsers.D2A/D2I files in the NowSMS directory.

SMSInDir=d:\path or SMSInDir=\\server\path

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

This setting can be applied in the [SMSGW] section of the SMSGW.INI file to specify the location of the SMS-IN directory which is used to queue received SMS messages that are pending for delivery to a 2-way command. By default this is the "SMS-IN" subdirectory of the NowSMS installation.

DataDir=d:\path or DataDir=\\server\path

This setting can be applied in the [MMSC] section of the MMSC.INI file to specify the location of the MMS Message Store for messages pending delivery. By default this is the "MMSCData" subdirectory of the NowSMS installation.

MMSDir=d:\path or MMSDir=\\server\path

This setting can be applied in the [MMSC] section of the MMSC.INI file to specify the location of the MMS-IN directory that is used when received MMS messages are converted to a file/ directory based interface. By default this is the "MMS-IN" subdirectory of the NowSMS installation.

MMSCUsersDir=d:\path or MMSCUsersDir=\\server\path

This setting can be applied in the [MMSC] section of the MMSC.INI file to specify the location of the "MMSC Users" database. By default this is the "MMSCUsers" subdirectory of the NowSMS installation, and the MMSCUsers.DB file in the NowSMS directory.

MMSSMSDataDir=d:\path or MMSSMSDataDir=\\server\path

This setting can be applied in the [MMSC] section of the MMSC.INI file to specify the location of the MMS Message Store for messages that have been converted to an "SMS with web link". By default this is the "MMSSMS" subdirectory of the NowSMS installation.

VASPConfigDir=d:\path or VASPConfigDir=\\server\path

This setting can be applied in the [MMSC] section of the MMSC.INI file to specify the location of all configuration information for incoming and outgoing MMSC routes (e.g., "MMSC VASP" and "MMSC Routing" definitions). By default this is the "VASPIN" and "VASPOUT" subdirectories of the NowSMS installation, along with the VASPIN.D2A/D2I and VASPOUT.D2A/D2I files.

VASPQDir=d:\path or VASPQDir=\\server\path

This setting can be applied in the [MMSC] section of the MMSC.INI file to specify the location of the MMS message queue for messages pending delivery to an external MMSC ("MMSC Routing"). By default this is the "VASPQ" subdirectory of the NowSMS installation.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Web Menu Interface

When the menu driven web interface is enabled, it is easy to test the ability of sending various types of SMS messages.

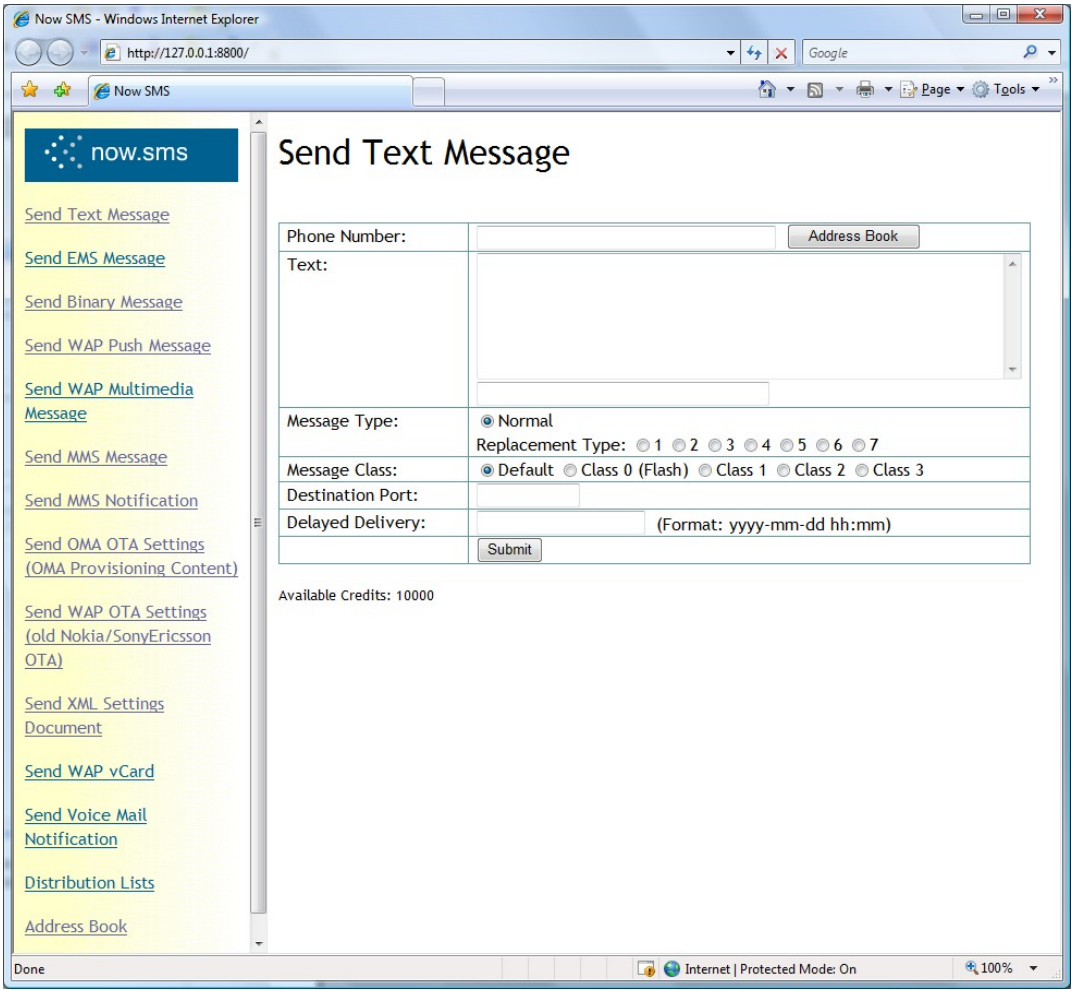
To enable the menu driven web interface of the gateway, you must check "Enable menu driven web interface" on the "Web" configuration dialog. When that option is enabled, you can connect to the web interface with a web browser. On the "Web" configuration dialog, there is a setting named "Port number for web interface". To connect to the web interface of the gateway, connect to http://ip.address:port, where "ip.address" is the IP address or host name of the PC running the gateway, and "port" is the port number specified for the web interface.

In a default configuration, the web menu interface can be accessed on the gateway PC by pointing a web browser to **http://127.0.0.1:8800**.

For more information on configuring the Web Menu Interface, see Configuring the Web Interface on page 60).

With a web browser, connect to the web port configured for the SMS gateway, and an interface similar to the following will be displayed:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

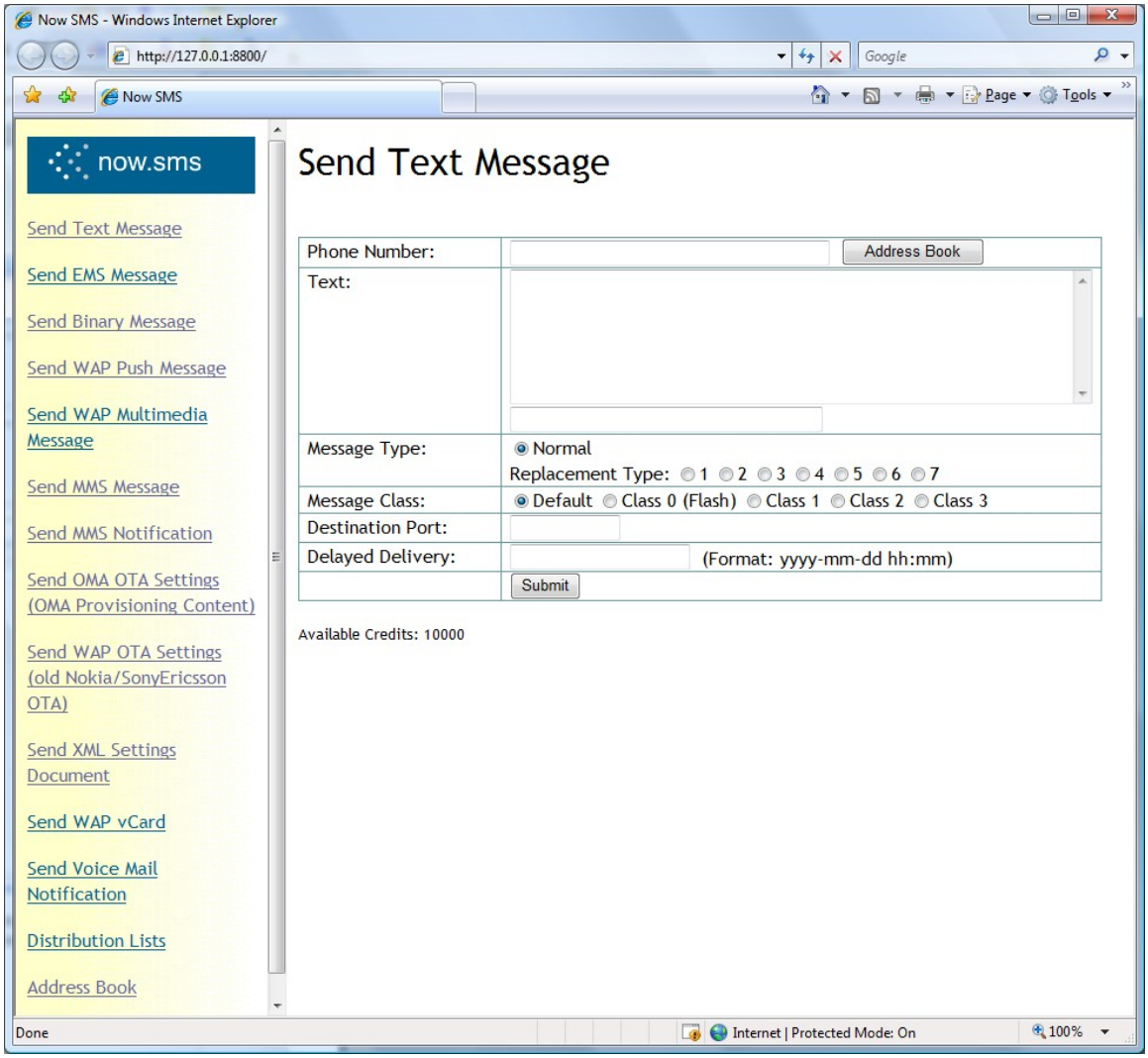


This web page provides a menu driven interface for sending various types of SMS and MMS messages. For information on how to send specific types of messages, please refer to the appropriate section below:

- Send Text Message *(page 75)*
- Send EM S Message *(page 77)*
- Send Binary Message *(page 84)*
- Send WAP Push Message *(page 90)*
- Send Multimedia Content Message *(page 98)*
- Send MMS Message *(page 94)*
- Send MMS Notification *(page 97)*
- Send OMA OTA Settings *(page 112)*
- Send WAP OTA Settings *(page 101)*
- Send XML Settings Document *(page 118)*
- Send Voice Mail Notification *(page 121)*

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send Text Message



To send a text message, simply enter a phone number and the text of your message. If the message is longer than 160 characters, the gateway will automatically use concatenated SMS ("long SMS") message support to send the entire message.

The "**Message Type**" would normally be set to "Normal". Setting a "Replacement Type" value means that if the gateway sends a subsequent message with the same replacement type value, this will replace any previous messages that were sent by the same sender with the same replacement type value.

When submitting an SMS message via URL parameters *(see page 187)*, replacement type values 1 thru 7 correspond to settings of PID=41 thru PID=47.

"**Message Class**" settings are generally used only for testing, except for "Class 0 (Flash)" messages which can be occasionally useful. A "flash" message is an SMS message that is

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

automatically opened on the display of the receiving phone, and is normally not saved to the phone's inbox, so that once the user exits the message, the message automatically disappears.

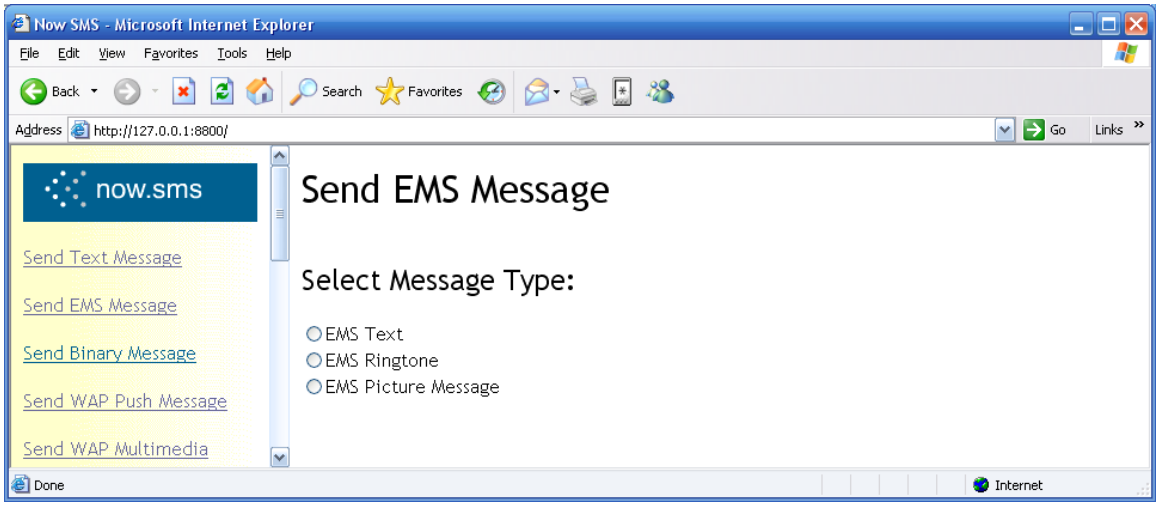
When submitting an SMS message via URL parameters *(see page 187)*, message class settings of 0 thru 3 correspond to settings of DCS=10 thru DCS=13. Note that NowSMS will automatically convert these DCS values if the message text contains characters that must be encoded using Unicode characters. However, some SMSC connections, such as SMPP, will not support flash messages that contain Unicode text.

"Destination Port" is useful when sending messages to a Java MIDlet running on a mobile phone. When submitting an SMS message via URL parameters, it is possible to use the "&DestPort=" parameter for specifying this setting.

"Delayed Delivery" allows a message to be submitted to NowSMS, but queued for processing at a future date and time. When submitting an SMS message via URL parameters, it is possible to use the "&DelayUntil=" parameter for specifying this setting.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send EMS Message



The Send EMS Message form contains some options to send some common types of EMS and Nokia Smart Messaging messages.

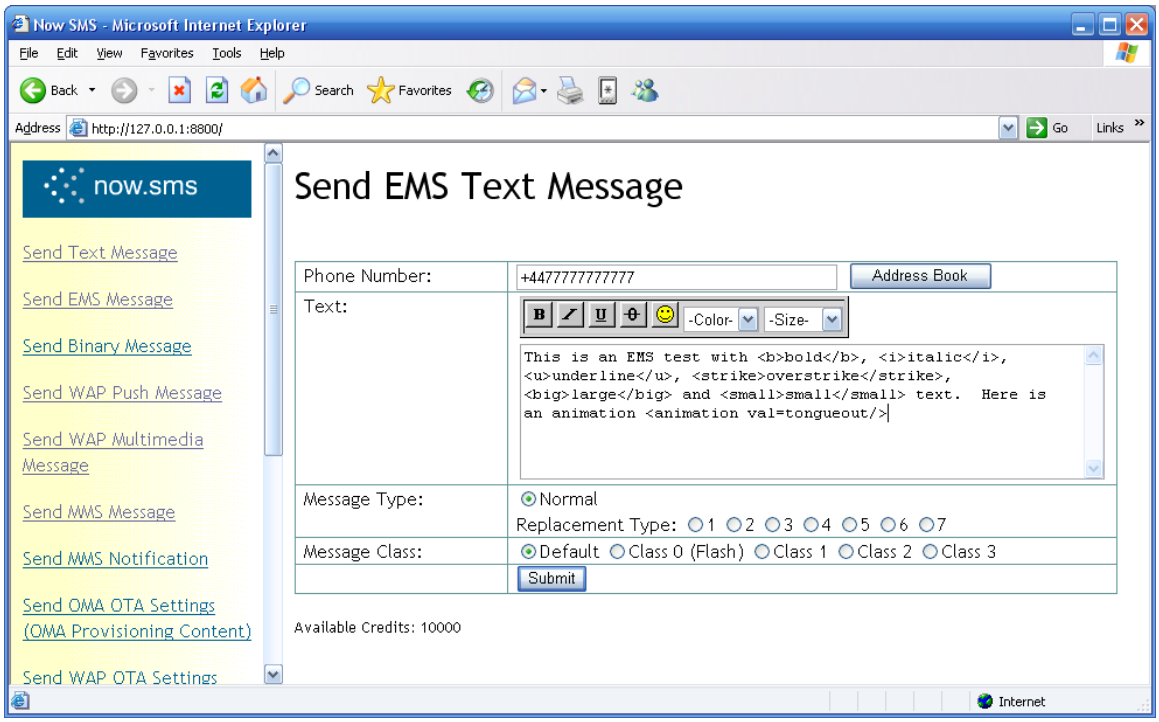
The "**EMS Text**" option allows you to send text messages that include EMS attributes for text formatting, such as bold, italics and large or small text. EMS text messages can also included predefined animations and sounds which are pre-loaded on EMS compatible phones. NowSMS also supports generating the EMS text formatting codes to specify colors to be used in text messages, however this functionality does not appear to be very widely supported in current handsets. EMS Text messages are an interesting messaging option, because the EMS standard is designed in such a way that an EMS text message is gracefully downgraded when delivered to a recipient handset that does not support EMS. If the recipient handset does not understand EMS, it will display only the text of the message.

The "**EMS ring tone**" option allows you to submit ring tone data in either RTTTL, iMelody or MIDI format, and send the ring tone out in either EMS, Nokia Smart Messaging, or WAP Push/MIDI format. These formats are largely supported for the sake of interfacing with older or less capable handsets, as newer handsets often support true tone formats based upon MP3.

The "**EMS Picture Message**" option allows you to send simple picture messages using either the EMS or Nokia Smart Messaging format. These formats refer to the old monochrome/black & white images that are sent via SMS, as opposed to the more advanced functionality offered by MMS. While there is little use for monochrome images in a Technicolor world, these message types can be useful for applications that need to send bar code types of images via SMS, where they cannot assume that recipient devices will have MMS or WAP capabilities.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send EMS Text Message



The "**EMS Text**" option allows you to send text messages that include EMS attributes for text formatting, such as bold, italics and large or small text. EMS text messages can also included predefined animations and sounds which are pre-loaded on EMS compatible phones. NowSMS also supports generating the EMS text formatting codes to specify colors to be used in text messages, however this functionality does not appear to be very widely supported in current handsets. EMS Text messages are an interesting messaging option, because the EMS standard is designed in such a way that an EMS text message is gracefully downgraded when delivered to a recipient handset that does not support EMS. If the recipient handset does not understand EMS, it will display only the text of the message.

The NowSMS web form includes a simple editor that inserts tags into the message which specify where text attributes should be changed, or where pre-defined animations should be inserted.

Turns on or off the **bold** text attribute. NowSMS inserts in the text to indicate the beginning of a bold section, and to indicate the end.

Turns on or off the *italic* text attribute. NowSMS inserts in the text to indicate the beginning of an italic section, and to indicate the end.

Turns on or off the underline text attribute. NowSMS inserts <u> in the text to indicate the beginning of an underline section, and </u> to indicate the end.

Turns on or off the ~~overstrike~~ text attribute. NowSMS inserts <strike> in the text to indicate the beginning of an overstrike section, and </strike> to indicate the end.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Displays a menu of pre-defined animations and sounds that can be inserted into the EMS message. These animations and sounds are defined as part of the EMS standard, and will vary slightly between different mobile phones.

Animations defined in the EMS standard include: Flirty, Glad, Skeptical, Sad, Wow, Crying, Winking, Laughing, Indifferent, In Love, Confused, Tongue Out, Angry, Glasses, and Devilish.

When an animation is inserted into the message, NowSMS inserts <animation val=xxxx/> to indicate the placement of the animation. xxxx is replaced with the name of the animation from the above list. Spaces are removed if present, for example <animation val=tongueout/> would indicate a placeholder for the "tongue out" animation.

Sounds defined in the EMS standard include: Chimes high, Chimes low, Ding, Ta Da, Notify, Drum, Claps, Fan Fare, Chords high, and Chords low.

When a sound is inserted into the message, NowSMS inserts <sound val=xxxx/> to indicate the placement of the animation. xxxx is replaced with the name of the sound from the above list. Spaces are removed if present, for example <sound val=tada/> would indicate a placeholder for the "ta da" sound.

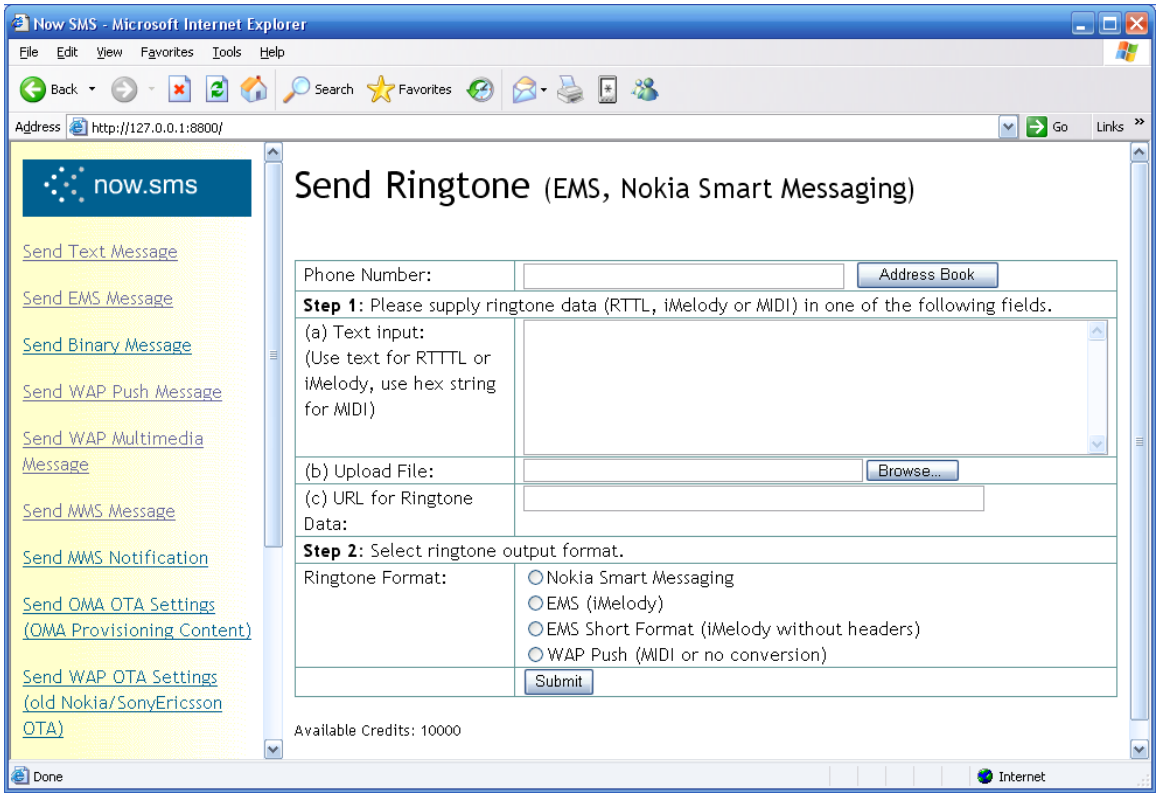
The "**Color**" drop-down allows an EMS text colour attribute to be specified. Colours supported in the EMS standard include: black, green, red, blue, yellow, purple (magenta), cyan, gray, and white.

When a colour attribute is inserted into the message, NowSMS inserts <color val=xxxx> to indicate the beginning of the block of coloured text, and </color> to mark the end of a block of coloured text. xxxx can be any of the colours listed above, or it can be a numeric value between 0 and 15 to indicate a colour code as defined in the EMS specification.

The "**Size**" drop-down allows attributes to be inserted to indicate small, normal, or large text. NowSMS inserts <small> to indicate the beginning of a section of small text, and </small> to mark the end of the section. NowSMS inserts <big> to indicate the beginning of a section of large text, and </big> to mark the end of the section. Normal text does not require an indicator. As an example, switching from large to small text would insert </large><small>, with </large> ending the large section of text, and <small> beginning the small section of text. Switching from large to normal text would insert </large>, with large ending the large section of text, implying that the text size returns to normal.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send EMS ring tone



The **"EMS ring tone"** option allows you to submit ring tone data in either RTTTL, iMelody or MIDI format, and send the ring tone out in either EMS, Nokia Smart Messaging, or WAP Push/MIDI format. These formats are largely supported for the sake of interfacing with older or less capable handsets, as newer handsets often support true tone formats based upon MP3.

To send a ring tone, you need to supply ring tone data. Ring tone data can be submitted either as text input, as a file to be uploaded, or via an http:// URL reference to a file that resides on a separate web server. The ring tone data must be in RTTTL, iMelody or MIDI format.

Now Mobile does not provide technical support on the creation or deployment of ring tone services. The limited conversion options provided in the Now SMS web interface are intended as a convenience. While NowSMS may be used for the delivery of ring tone content, we strongly recommend that you evaluate other software packages to aid in the creation and conversion of ring tones.

Now Mobile also does not provide technical support or guidance regarding which ring tone formats are supported by which mobile phone models. Ring tone delivery can be a complex business, and the NowSMS product is focused on message delivery, not ring tone authoring.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

NowSMS supports the following input ring tone formats:

1.) **RTTTL** is the ring tone format that is used in the Nokia Smart Messaging standard. Here is a simple RTTTL example featuring the opening theme of Beethoven's Fifth Symphony:

```
fifth:d=4,o=5,b=63:8P,8G5,8G5,8G5,2D#5
```

2.) **iMelody** is the ring tone format that is used in the EMS standard. Here is a simple iMelody example featuring the opening them of Beethoven's Fifth Symphony:

```
BEGIN:IMELODY
NAME:fifth
BEAT:63
STYLE:S0
MELODY:r3*3g3*3g3*3g3*3#d1
END:IMELODY
```

3.) **MIDI** is a slightly more capable ring tone format which uses a binary file format instead of a text format. While it is possible to convert a small MIDI file into a text string of hex characters, more commonly, a MIDI file would either be uploaded to the NowSMS server, or referenced via URL from another web server.

4.) It is also possible to use this web form to send ring tones of other formats out via a WAP Multimedia Message *(see page 98)*. When a ring tone file in a format other than RTTTL, iMelody or MIDI is sent via this web form, it cannot be submitted as text input, and needs to be submitted as a file to be uploaded, or via an http:// URL reference to a file that resides on a separate web server. In this case, the output ring tone format must be "WAP Push", and NowSMS will send the ring tone out via WAP Multimedia Push without performing any conversion of the ring tone data.

NowSMS supports the following ring tone output formats:

1.) **Nokia Smart Messaging** – This is the binary encoding for RTTTL, which was originally defined by Nokia. (Note that NowSMS currently only supports the sending these messages out in binary format. The text "//SCKL" format may be supported in a future release.)

2.) **EMS (iMelody)** – The ring tone is converted to iMelody, if necessary, and encoded as an EMS message.

3.) **EMS Short Format (iMelody without headers)** – The ring tone is converted to iMelody, if necessary. As EMS can be rather verbose, the headers are stripped from the iMelody data, and it is then encoded as an EMS message.

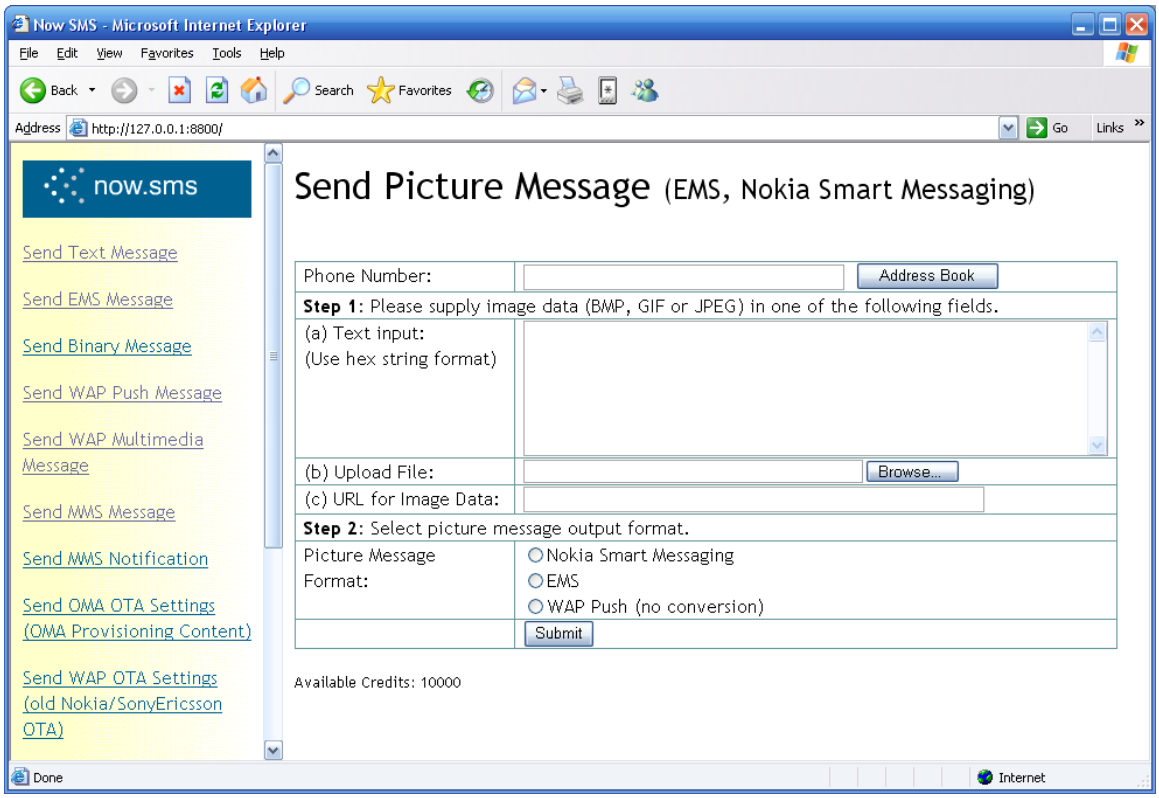
EMS iMelody messages are typically larger than Nokia Smart Messaging encodings because of the verbose iMelody headers. It is therefore more likely that a longer melody will be forced to span multiple SMS messages. Many EMS compatible phones do not support melodies that span multiple SMS messages, requiring the use of the EMS Short Format to attempt to fit the ring tone into a single message.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

4.) **WAP Push (MIDI or no conversion)** – If the input ring tone is in RTTTL or iMelody format, it is converted to MIDI. Otherwise, no conversion is performed. The output ring tone is delivered as a WAP Multimedia Message *(see page 98)*.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send EMS Picture Message



The **"EMS Picture Message"** option allows you to send simple picture messages using either the EMS or Nokia Smart Messaging format. These formats refer to the old monochrome/black & white images that are sent via SMS, as opposed to the more advanced functionality offered by MMS. While there is little use for monochrome images in a Technicolor world, these message types can be useful for applications that need to send bar code types of images via SMS, where they cannot assume that recipient devices will have MMS or WAP capabilities.

To send a picture message , you need to supply picture or image data. Image data can be submitted either as text input, as a file to be uploaded, or via an http:// URL reference to a file that resides on a separate web server. The image data must be in BMP, GIF or JPEG format. (To input a BMP, GIF or JPEG image as a text string, it must be converted to a text string of hex characters where each binary byte of the image is represented as two hex characters. File upload or referencing a web server URL that contains the image is usually easier.) Input images should have a width in pixels that is a multiple of 8.

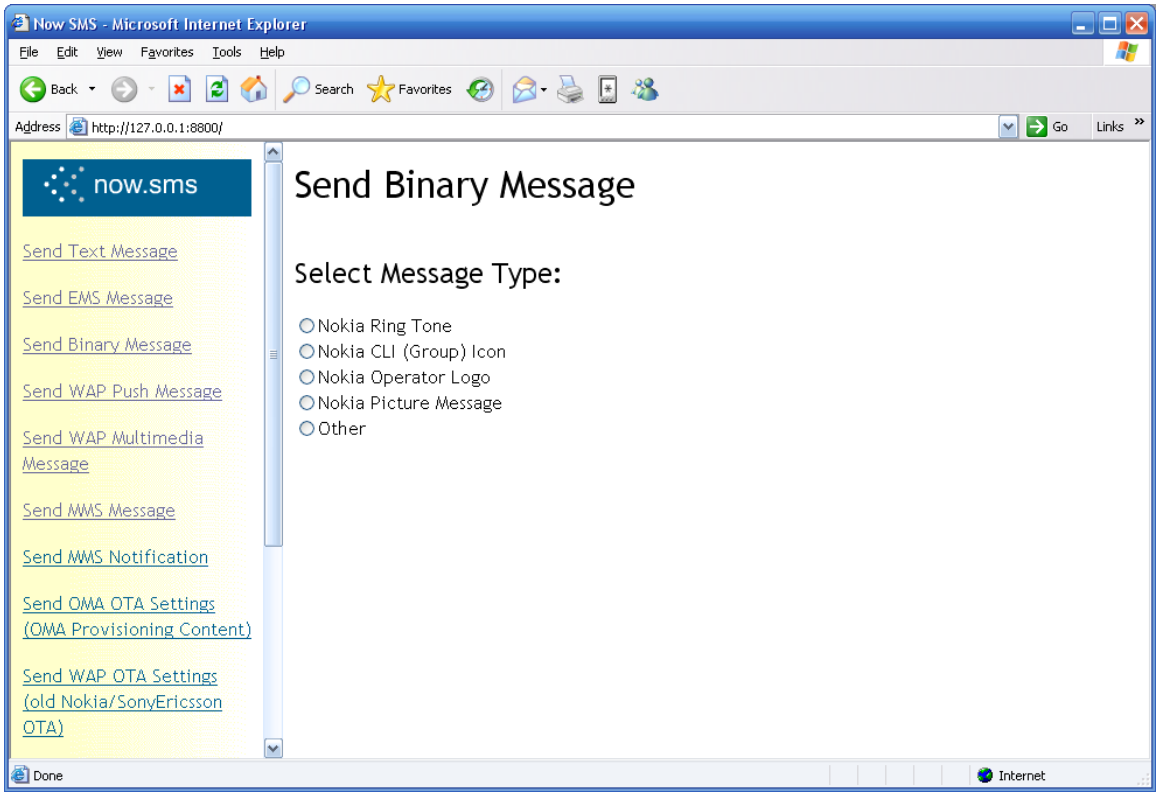
NowSMS supports the following picture message output formats from this interface:

1.) Nokia Smart Messaging
2.) EMS
3.) WAP Multimedia Message *(see page 98)*

Keep in mind that images sent via this interface that are to be converted to Nokia Smart Messaging or EMS message should be kept small in size. For larger images, use MMS.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

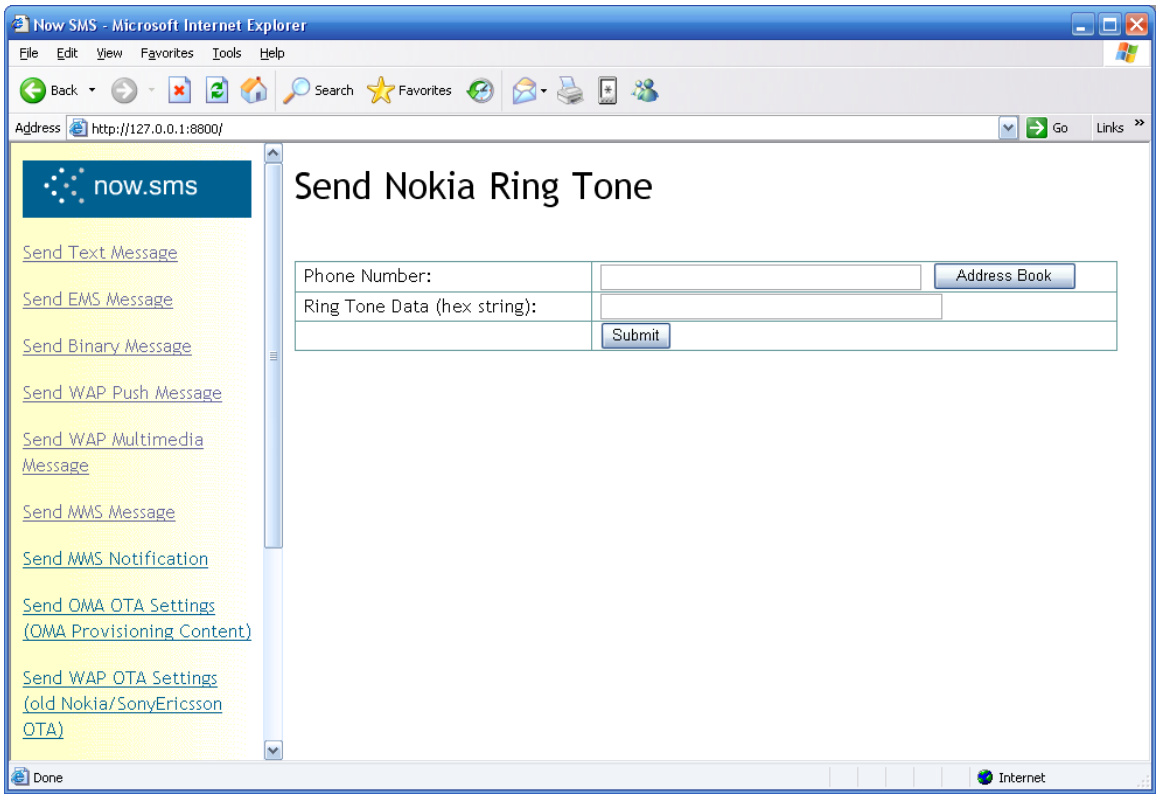
Send Binary Message



Sending a binary message through the web interface typically requires more knowledge of the binary SMS protocol that you are attempting to use. HTML forms are included for simplifying the process of sending Nokia Smart Messaging types, along with a general form for sending any binary message. Please note that additional Nokia Smart Messaging functionality is also provided by the **Send EMS Message** web form *(see page 77)*.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send Nokia Ring Tone



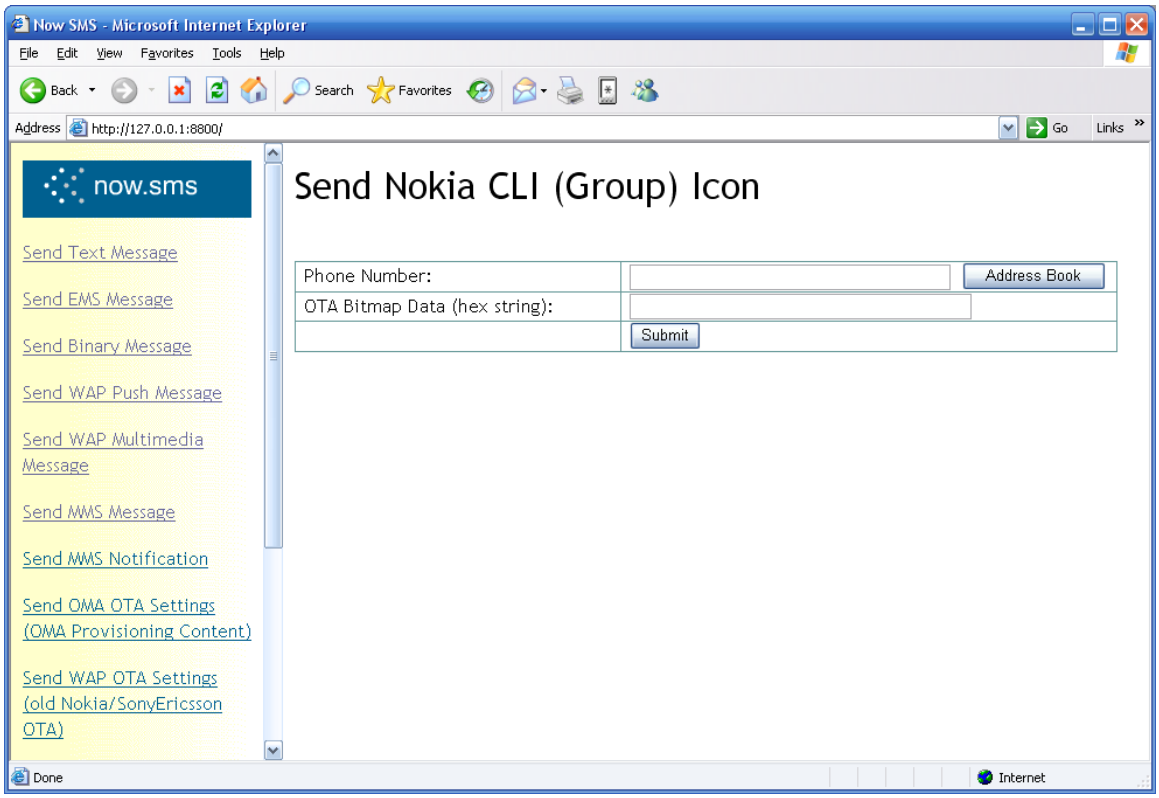
To send a Nokia ring tone, you must have a hex string value for the ring tone data. The hex string format represents two characters for each binary byte of ring tone data. Documentation of the ring tone data format is beyond the scope of this document.

For those who wish to send ring tones programmatically via the Now SMS/MMS Gateway, note that this form includes the following hidden fields which are included as URL parameters when submitting the message to the server:
UDH = 06050415811581
PID = 0
DCS = F5

Please note that additional Nokia Smart Messaging functionality is also provided by the **Send EMS Message** web form *(see page 77)*.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send Nokia CLI (Group) Icon



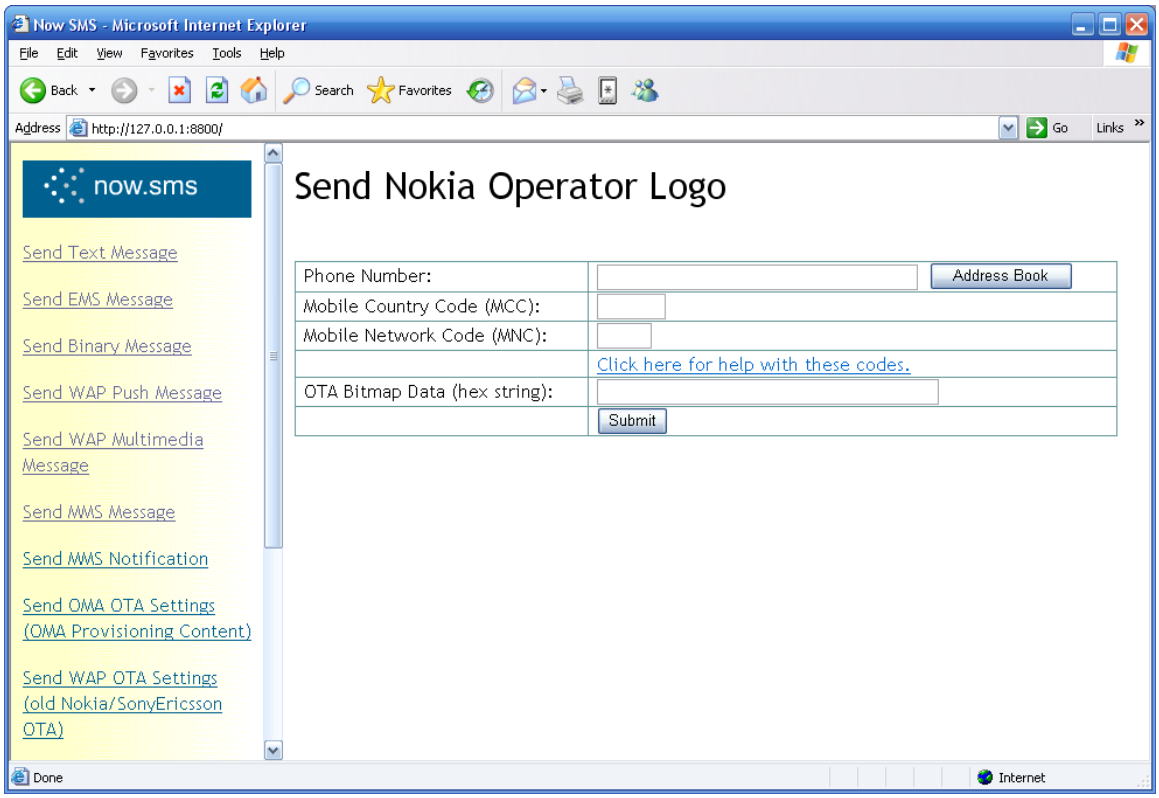
To send a Nokia Group Icon, you must have a hex string value for an OTA Bitmap, as defined by the Nokia Smart Messaging specification. The hex string format represents two characters for each binary byte of OTA Bitmap data. Documentation of the OTA Bitmap data format is beyond the scope of this document.

For those who wish to send Nokia Group icons programmatically via the Now SMS/MMS Gateway, note that this form includes the following hidden fields which are included as URL parameters when submitting the message to the server:
UDH = 06050415831583
PID = 0
DCS = F5

JavaScript in the HTML form adds the hex string "30" to the beginning of the OTA Bitmap string and submits it as the "Data" parameter in the URL.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send Nokia Operator Logo



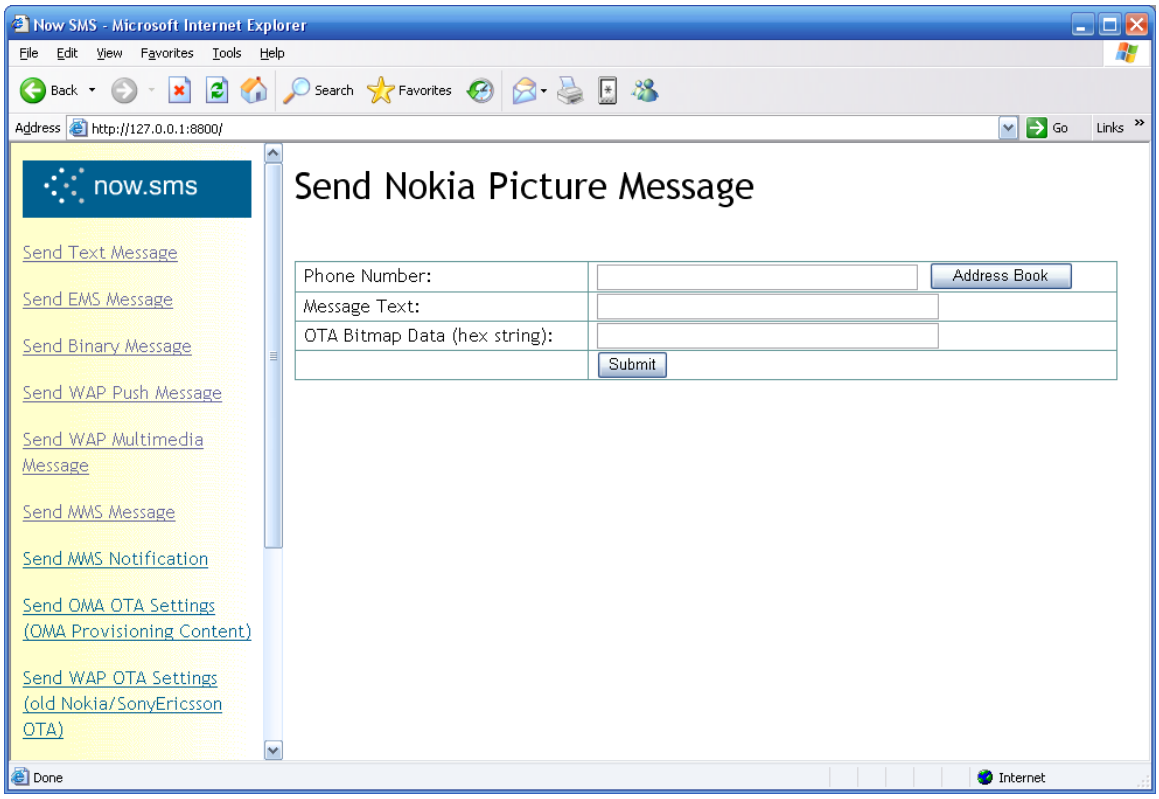
Nokia Operator logos are one of the more complicated of the Nokia Smart Messaging formats. To send a Nokia Operator logo, you must have a hex string value for an OTA Bitmap, as defined by the Nokia Smart Messaging specification. The hex string format represents two characters for each binary byte of OTA Bitmap data. Documentation of the OTA Bitmap data format is beyond the scope of this document. You must also know the Mobile Country Code (MCC) and Mobile Network Code (MNC) values of the network operator to which the recipient is subscribed. A link on the form provides more information on MCC and MNC codes, and a pointer to the URL http://www.gsmworld.com/roaming/gsminfo/index.shtml, from which you can look up the MCC and MNC codes of various network operators.

For those who wish to send Nokia Operator logos programmatically via the Now SMS/MMS Gateway, note that this form includes the following hidden fields which are included as URL parameters when submitting the message to the server:
UDH = 06050415821582
PID = 0
DCS = F5

JavaScript in the HTML converts the MCC and MNC codes into the format required by the Nokia Smart Messaging specification, and combines them with the OTA Bitmap data to create a valid operator logo message in the URL "Data" parameter submitted by the form.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send Nokia Picture Message



Nokia Picture Messaging should not be confused with MMS picture messaging. The Nokia picture messaging format typically only allows for the submission of small specially formatted black and white pictures, whereas MMS provides support for larger color images in a variety of different formats.

To send a Nokia Picture Message, you must have a hex string value for an OTA Bitmap, as defined by the Nokia Smart Messaging specification. The hex string format represents two characters for each binary byte of OTA Bitmap data. Documentation of the OTA Bitmap data format is beyond the scope of this document. A picture message also includes a short text message.

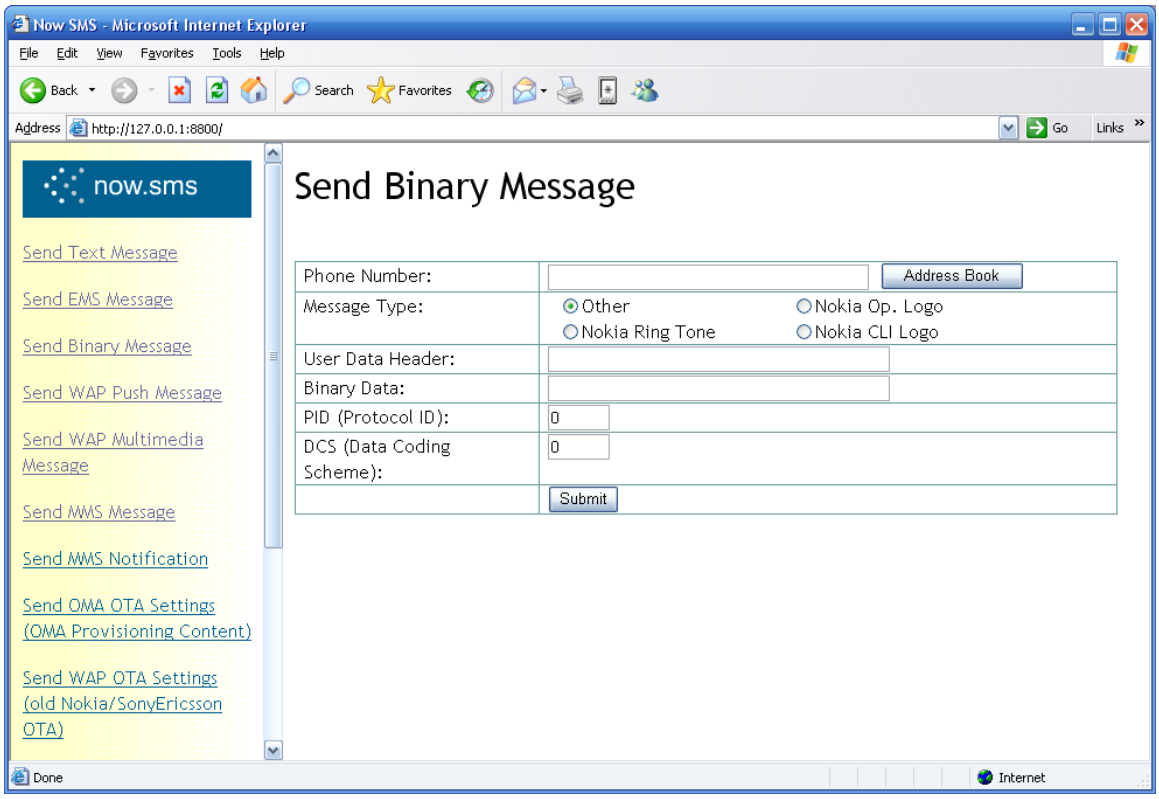
For those who wish to send Nokia Picture Messages programmatically via the Now SMS/MMS Gateway, note that this form includes the following hidden fields which are included as URL parameters when submitting the message to the server:
UDH = 060504158A158A
PID = 0
DCS = F5

JavaScript in the HTML form combines the message text and the OTA bitmap data to create a valid picture message in the URL "Data" parameter submitted by the form.

Please note that additional Nokia Smart Messaging functionality is also provided by the **Send EMS Message** web form *(see page 77)*.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

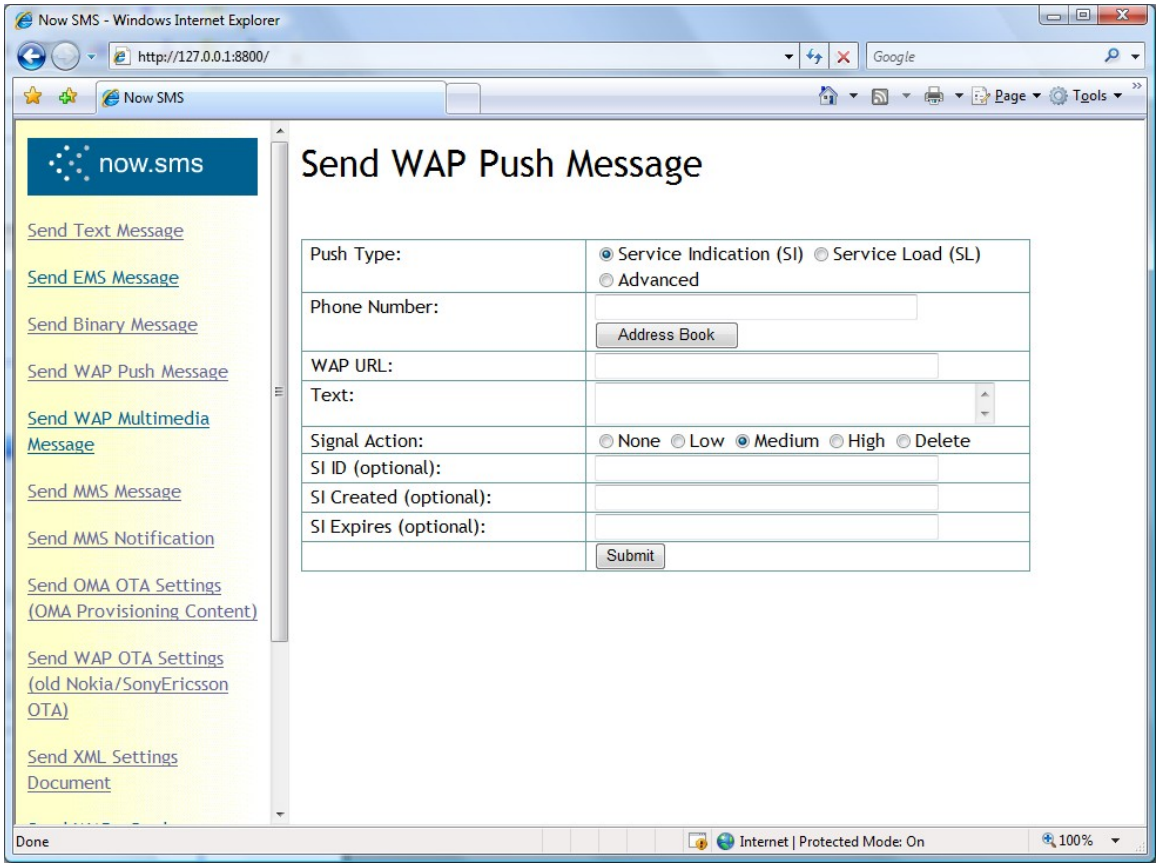
Send Binary Message Other



The "Send Binary Message Other" form allows for the submission of other types of binary messages. This typically requires more knowledge of the binary SMS protocol that you are attempting to use, but this web form can be convenient for testing.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send WAP Push Message



This web form provides a simple interface for sending a WAP Push message.

Two basic types of WAP Push messages are supported from the web interface, Service Indication (SI) and Service Load (SL). The menu will change slightly depending upon whether Service Indication or Service Load has been selected as the "**Connection Type**".

The web interface also provides an "Advanced" option for generating WAP Push messages of any content type, including custom WAP Push WSP headers, which is useful for testing and experimentation.

A **Service Indication** message is what we think of as a standard WAP Push message. The push basically consists of some text and a URL. When the push is received, the mobile phone shows the text, along with an option to load the URL.

A **Service Load** message is a type of WAP Push that was designed for system applications. The push does not allow any text to be sent, only a URL. The original design intent was that the mobile phone would automatically open the URL without user intervention. Of course, this presents significant security concerns, so many mobile phones either do not support Service Load, or they treat it the same as a Service Indication message, with the disadvantage that the push cannot include any text to identify it. Frequently this type of

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

message results in a message on the phone indicating "Service Message Received", with an option asking if you wish to load the URL, and no further information. We recommend using the Service Load type of push only for specialised applications, while the Service Indication type is for general use.

"**Phone Number**" specifies a comma delimited list of recipients to receive the message.

"**WAP URL**" specifies the HTTP URL to be associated with the push. (If the user chooses to load the push message, they will be connected to this URL.) If the http:// prefix is not included, it will be assumed.

"**Text**" specifies some text to be displayed when the push message is first opened. This text is typically displayed along with a link to load the URL associated with the push.

"**Signal Action**" can specify a different type of alert to be associated with the push. While this is not very widely supported, the general intent is to associate a priority with the alert, so that the device might take greater action to alert the user to a "High" priority push, as opposed to a "Low" or "Medium" priority push.

Of particular interest in the "Signal Action" options is the "Delete" action. If a push has an "SI ID" associated with it, it is possible to later send a "Signal Action = Delete" push with the same "SI ID" to delete the previous push message from the device inbox.

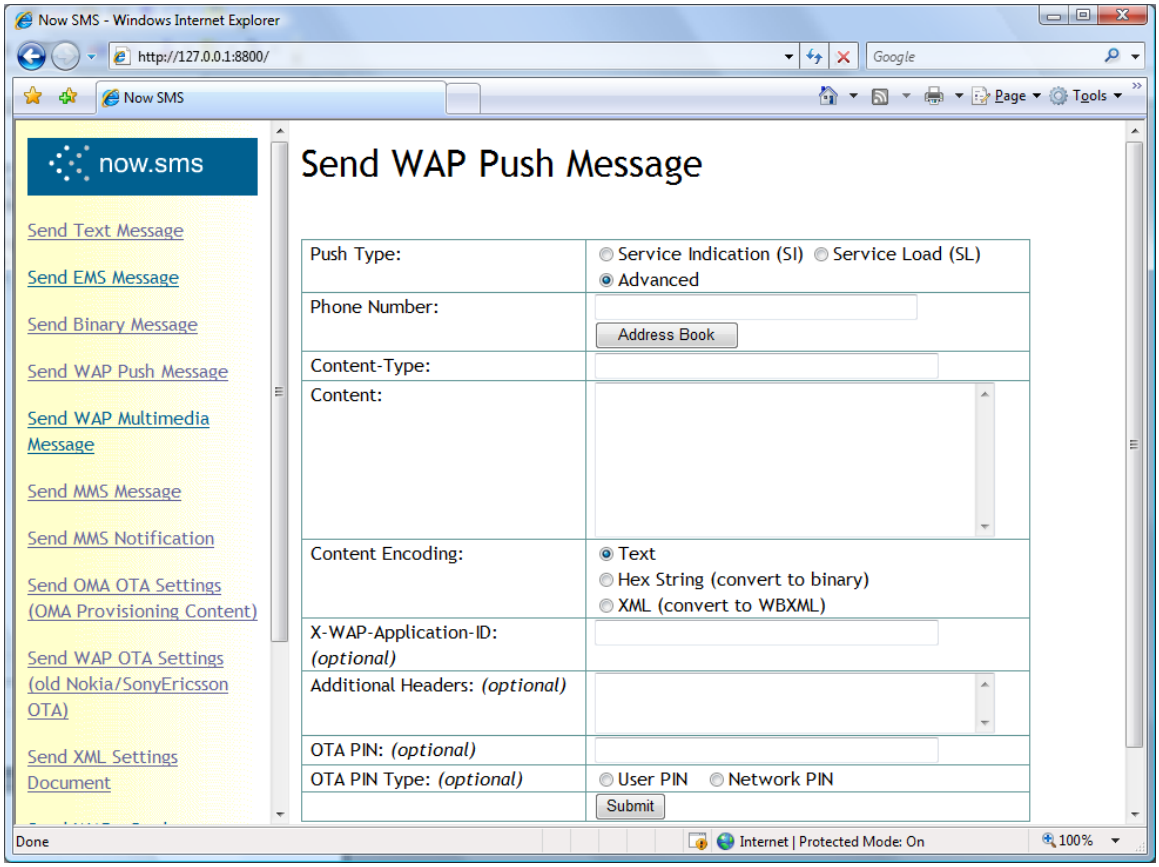
Similarly, if a mobile device receives a push message with an "SI ID" that matches that of a previously received push that is still in its inbox, the new push message should replace the existing push message.

The "**SI Created**" field specifies a creation date/time stamp to be associated with the push. If specified, this date/time stamp should take the format "yyyy-mm-ddThh:mm:ssZ", specifying a date/time value relative to GMT. For example, "2006-02-24T00:00:00Z".

The "**SI Expires**" field specifies a date/time at which the receiving device should automatically expire the push. This is a date/time value relative to GMT, in the format "yyyy-mm-ddThh:mm:ssZ". For example, "2006-02-24T00:00:00Z".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send WAP Push Advanced



The "Advanced" WAP Push form allows for the sending of WAP Push messages of any content type, and supports the ability to include custom WAP Push WSP headers.

The "**Content-Type**" field should contain the MIME content type of the content to be pushed.

The "**Content**" field should contain the content of the push. The content can either be encoded as plain text, or for binary content types the content can be represented as a string of hexadecimal characters.

The "**Content Encoding**" field specifies whether the content is plain text or a string of hexadecimal characters. If you are sending one of the content types supported by "Send XML Settings", but wish to include extra headers, it is also possible for NowSMS to perform XML to WBXML conversion for any of the XML content types supported by "Send XML Settings" (see page 118). If you wish to have NowSMS perform XML to WBXML conversion, select this conversion option for "Content Encoding".

The "**X-WAP-Application-ID:**" field can contain a numeric identifier or text string to indicate a destination WAP Push application id.

The "**Additional Headers**" field can contain a list of additional headers to be included in the WAP Push WSP header. Each header should be on a separate line, with a format of

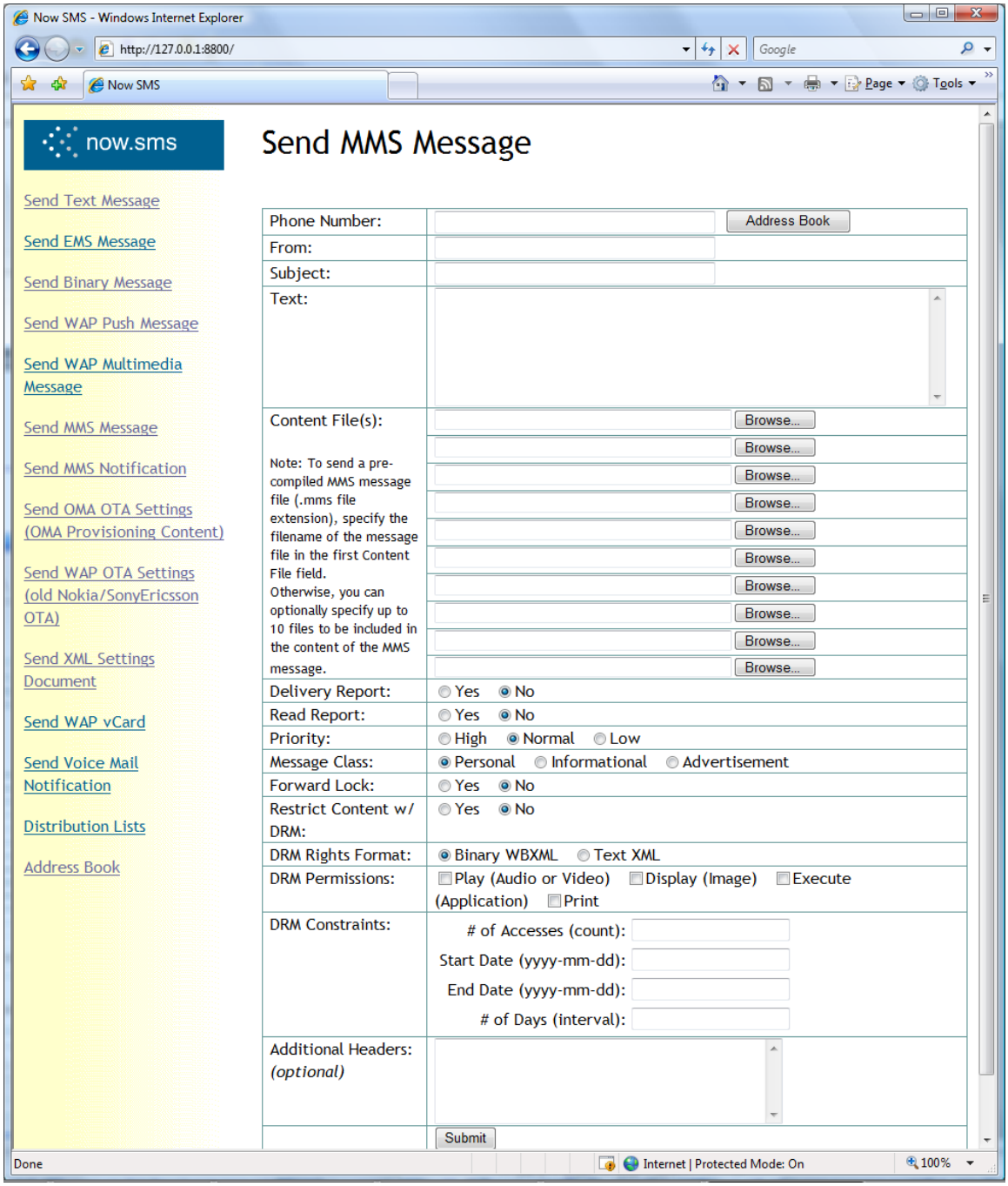
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Header-Name: Header-Value. As an alternative to text encoding, WSP headers can be specified directly using with hex string encoding. When NowSMS encounters a line of hex characters in the "Additional Headers" field, it assumes that this a pre-encoded WSP header value, and the binary equivalent of the hex string is included in the WSP header without any validation.

"**OTA PIN**" and "**OTA PIN Type**" can also be optionally specified to allow the push message to be signed.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send MMS Message



The "**Send MMS Message**" web form allows you to define a subject, message text, and optionally include multiple content files (uploaded via the browser). Content files may include text files, audio files, image files, SMIL files, and/or other supported MMS content types. The gateway automatically compiles the MMS message file and uses the gateway's built-in MMSC to send the message. An MMS message sent via this facility will be routed

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

via the MMSC, and could either be delivered directly by the MMSC, or could be routed by the MMSC to an operator MMSC for delivery.

Note that this menu interface also allows for the sending of a pre-compiled MMS message file. If you are sending a pre-compiled MMS message file, that file should be submitted as the only content file for the message, and it should have a ".mms" file extension.

The web form allows you set some message attributes, and to specify Digital Rights Management (DRM) restrictions over the content of the message.

"**Delivery Report**" specifies whether or not a delivery report is requested for the message. Note that any delivery report would be directed back to the phone number or e-mail address specified in the "From" address.

"**Read Report**" specifies whether or not a read receipt is requested for the message. Note that the receiving client may choose not to send a read receipt. Any read receipt report would be directed back to the phone number or e-mail address specified in the "From" address.

"**Priority**" is a user defined priority to be associated with the message. Generally, any priority definition associated with the message is ignored by the underlying transport, but the receiving client may decide to display messages differently based upon this priority setting.

"**Message Class**" is an attribute defined in the MMS specifications. "Personal" is the message type that is used for standard user-to-user communications.

Digital Rights Management Options

When sending an MMS message via this interface, it is possible to specify Digital Rights Management (DRM) restrictions over the content of the message.

The most basic level of DRM is forward locking. When "**Forward Lock**" is set to "Yes", this indicates that the receiving device should not allow any non-text objects in the message to be forwarded off of the device. The device may allow the user to extract pictures, videos or sounds from the message and save them on the phone. However, any such objects remain forward locked, such that they cannot be forwarded to another user or transferred to another device.

More advanced DRM restrictions can be applied to limit the number of times that the user can access an object, or start and end dates can be specified to limit how long the user can access an object.

These advanced DRM restrictions can be applied by setting "**Restrict Content w/ DRM**" to "Yes". When this setting is enabled, forward lock is also implied, and the value of the "Forward Lock" setting is ignored.

"**DRM Rights Format**" specifies whether the DRM rights object should be encoded using a Text XML format, or a Binary WBXML format. While Binary WBXML format is always preferred for separate delivery objects, many Nokia phones only support Text XML format for the combined delivery process supported by this interface.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"**DRM Permissions**" specify what types of access are allowed against the object. For example, an audio or video object requires "**Play**" permission before the user can access it. An image requires "**Display**" permission before the user can access it, and it requires "**Print**" permission if the user is to be allowed to print the image to a printer , perhaps over Bluetooth. An application requires "**Execute**" permission before the user can make use of the application. In all cases, the forward locking is assumed, so that the user is not allowed to forward or transfer the object from the device.

If you are sending multiple types of objects in the MMS message, check all permissions that are required for the different types of objects that you are sending.

"**DRM Constraints**" specify constraints with regard to how long the object should remain accessible to the user. It is possible to specify one or more of these constraints.

"**# of Accesses (count)**" specifies the the user can only access the object this number of times before access is no longer allowed.

"**Start Date (yyyy-mm-dd)**" specifies that the user will not be allowed to access the object until on or after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2006-02-24.)

"**End Date (yyyy-mm-dd)**" specifies that the user will not be allowed to access the object after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2006-02-24.)

"**# of Days (interval)**" specifies that the user will be allowed to access the object for this number of days after initial receipt of the object. The user can either enter a number of days here, or they can enter any valid value defined for the "<interval>" element in the OMA DRM Rights Expression Language specification. For example, P2Y10M15DT10H30M20S represents a duration of 2 years, 10 months, 15 days, 10 hours, 30 minutes and 20 seconds.

For additional information on DRM Restrictions, please see **Digital Rights Management** on page 320.

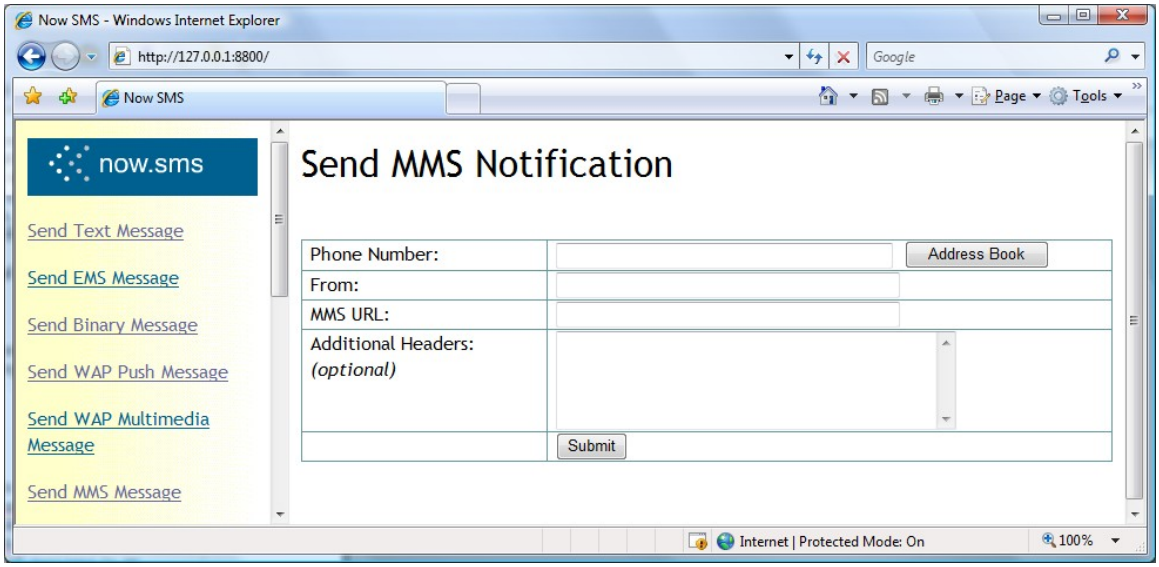
The "**Additional Headers**" option allows additional headers defined by the MMS Encapsulation Specification (MM1) to be inserted into the resulting MMS message. These MMS headers should be specified in a text format, such as:

```
X-Mms-MMS-Version: 1.3
X-Mms-Replace-ID: 20070524/12/ABCDEF01@mmsc
```

Note that NowSMS will filter headers that are actually included in the MMS message based upon the resulting MMS message and delivery method. When NowSMS is configured for direct delivery as an MMSC, it is possible to specify any MMS headers that are valid for either an M-Notification.ind or M-Retrieve.conf PDU, and NowSMS will include the headers in the resulting PDUs as appropriate. When NowSMS is routing MMS messages to an operator MMSC via a GPRS modem, it is possible to specify any MMS headers that are valid for an M-Send.req PDU.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send MMS Notification



The alternative MMS interface, "Send MMS Notification" is intended for more advanced developers, and configurations where it is possible for NowSMS to be used as an MMSC for direct delivery (although technically this interface bypasses the NowSMS MMSC). After creating a binary MMS message file, and storing the message file on a web server with a MIME type of "application/vnd.wap.mms-message", this dialog shows how the gateway can be used to send a notification to the message recipient that instructs the recipient's phone to connect to the specified URL to retrieve the MMS message content.

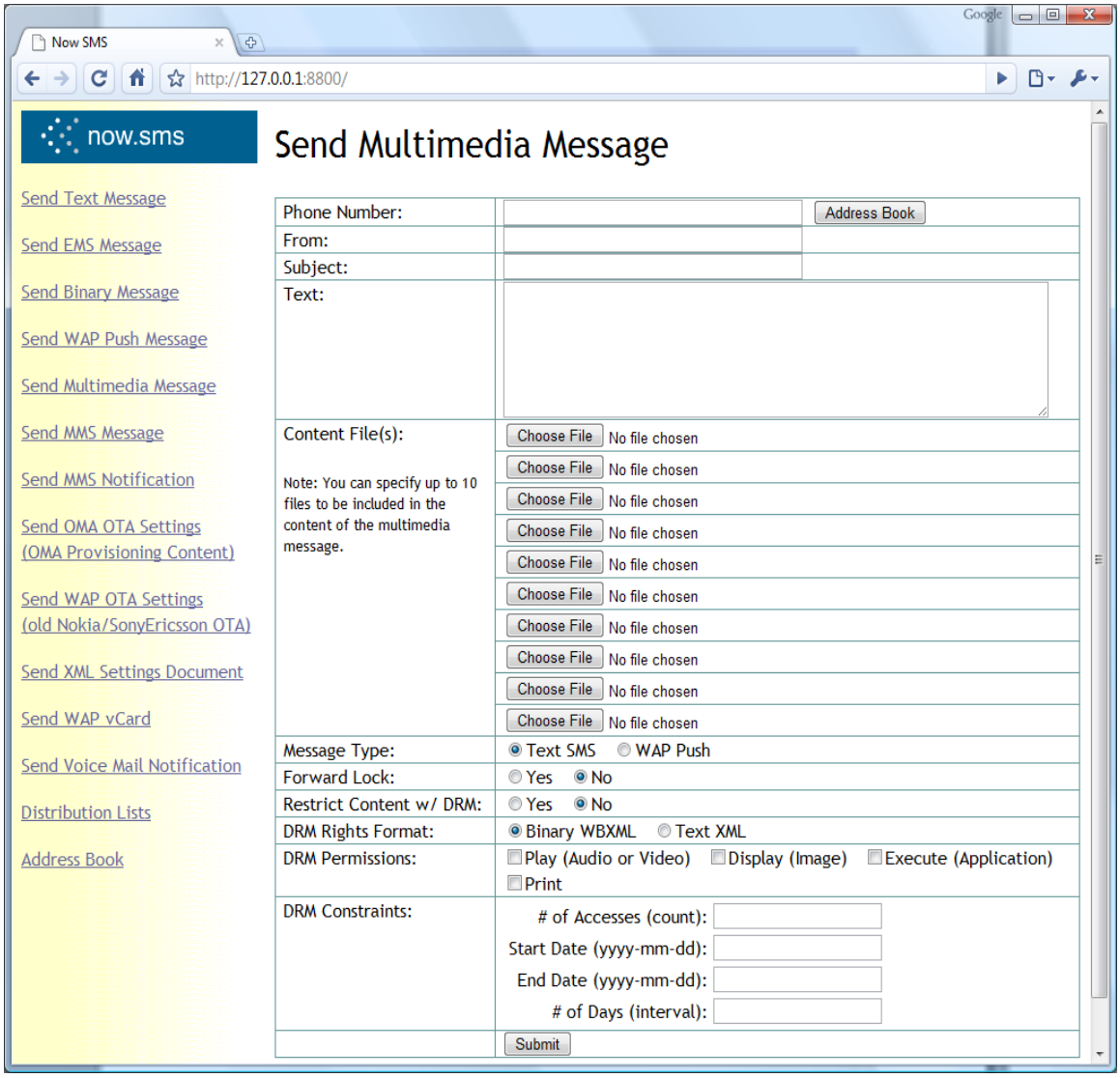
Please note that many mobile operator MMSCs block MMS delivery from external MMSCs, and it may not be possible for you phone to retrieve an MMS message from your own web server URL without changing settings on your mobile phone.

Note: When the "Send MMS Notification" function is used, the MMS Notification is sent to the recipient independent of the MMSC built-in to the gateway. The message recipient will fetch the message directly from the URL specified. As the message is not routed through the MMSC, the MMSC cannot provide dynamic content adaptation and conversion services.

The **"Additional Headers"** option allows additional headers defined by the MMS Encapsulation Specification (MM1) to be inserted into the resulting MMS notification message. By default, NowSMS assumes that it is sending an M-Notification.ind PDU (MMS Notification) for the specified URL. However, the "MMS URL" field can be left blank, and a different PDU type can be specified via the "X-Mms-Message-Type:" header (for example, "m-delivery-ind", "m-read-rec-ind" or "m-cancel-req").

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send Multimedia Content Message



The "Send Multimedia Message" option functions similar to the "Send MMS Message" option. It allows you to define a subject, message text, and optionally include multiple content files (uploaded via the browser). Content files may include text files, audio files, image files, Java applets, and/or other content types that might be supported by the receiving device. NowSMS automatically formats the content so that it is accessible via a dynamically generated URL on the MMSC, and sends either a WAP Push message, or a standard text SMS message, which includes a URL link to allow the content to be retrieved by the receiving device.

Use the **"Message Type"** setting to specify whether the link should be sent via a "Text SMS" message or a "WAP Push" message.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

When sending a Multimedia Message via this interface, it is possible to specify Digital Rights Management (DRM) restrictions over the content of the message, if DRM restrictions are supported by the receiving device.

The most basic level of DRM is forward locking. When "**Forward Lock**" is set to "Yes", this indicates that the receiving device should not allow any non-text objects in the message to be forwarded off of the device. The device may allow the user to extract pictures, videos or sounds from the message and save them on the phone. However, any such objects remain forward locked, such that they cannot be forwarded to another user or transferred to another device.

More advanced DRM restrictions can be applied to limit the number of times that the user can access an object, or start and end dates can be specified to limit how long the user can access an object.

These advanced DRM restrictions can be applied by setting "**Restrict Content w/ DRM**" to "Yes". When this setting is enabled, forward lock is also implied, and the value of the "Forward Lock" setting is ignored.

"**DRM Rights Format**" specifies whether the DRM rights object should be encoded using a Text XML format, or a Binary WBXML format. While Binary WBXML format is always preferred for separate delivery objects, many Nokia phones only support Text XML format for the combined delivery process supported by this interface.

"**DRM Permissions**" specify what types of access are allowed against the object. For example, an audio or video object requires "**Play**" permission before the user can access it. An image requires "**Display**" permission before the user can access it, and it requires "**Print**" permission if the user is to be allowed to print the image to a printer , perhaps over Bluetooth. An application requires "**Execute**" permission before the user can make use of the application. In all cases, the forward locking is assumed, so that the user is not allowed to forward or transfer the object from the device.

If you are sending multiple types of objects in the MMS message, check all permissions that are required for the different types of objects that you are sending.

"**DRM Constraints**" specify constraints with regard to how long the object should remain accessible to the user. It is possible to specify one or more of these constraints.

"**# of Accesses (count)**" specifies the the user can only access the object this number of times before access is no longer allowed.

"**Start Date (yyyy-mm-dd)**" specifies that the user will not be allowed to access the object until on or after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2006-02-24.)

"**End Date (yyyy-mm-dd)**" specifies that the user will not be allowed to access the object after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2006-02-24.)

"**# of Days (interval)**" specifies that the user will be allowed to access the object for this number of days after initial receipt of the object. The user can either enter a number of days here, or they can enter any valid value defined for the "<interval>" element in the

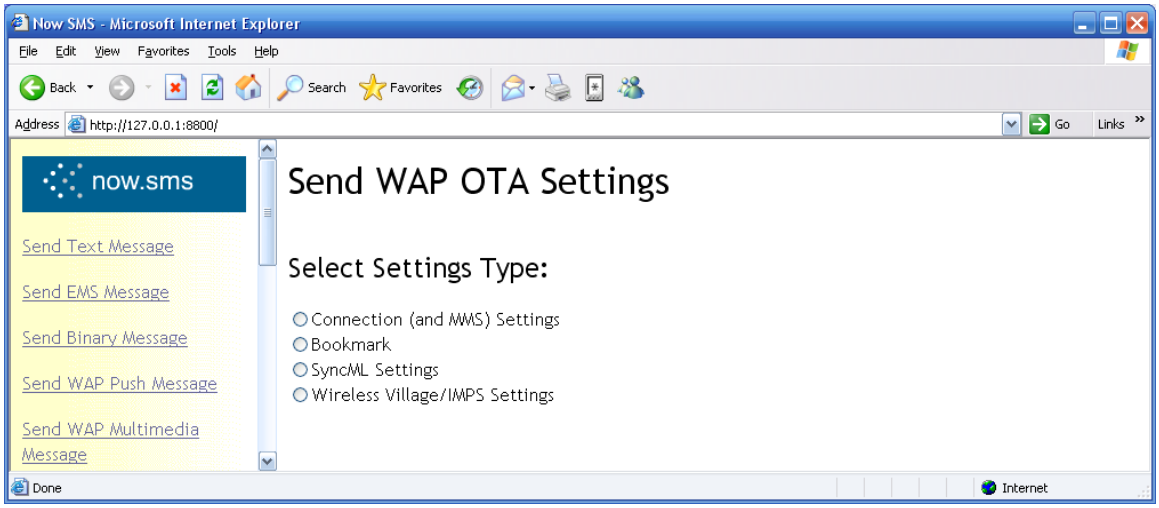
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

OMA DRM Rights Expression Language specification. For example, P2Y10M15DT10H30M20S represents a duration of 2 years, 10 months, 15 days, 10 hours, 30 minutes and 20 seconds.

For additional information on DRM Restrictions, please see **Digital Rights Management** on page 320.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send WAP OTA Settings



NowSMS supports sending OTA (Over-the-Air) configuration information to mobile devices in order to configure some phone settings, such as internet browser, MMS, SyncML, and instant messaging settings.

The "Send WAP OTA Settings" option sends configuration settings in a format that is compatible with the Nokia/Ericsson OTA Specification *(up to and including v7.1)*. This specification is mostly used by older devices, while most newer devices use the OMA (Open Mobile Alliance) OTA Provisioning Content, which is accessible via the **Send OMA OTA Settings** menu, which is described on page 112.

There are four different types of WAP OTA settings.

1) Connection (and MMS Settings) – This sends configuration settings for the WAP/internet browser and/or MMS client.
2) Bookmark – This sends a single bookmark for a web site to the mobile device (mostly supported only by older Nokia phones)
3) SyncML Settings – This sends configuration settings for the SyncML client
4) Wireless Village/IMPS Settings – This sends configuration settings for the Instant Messaging client.

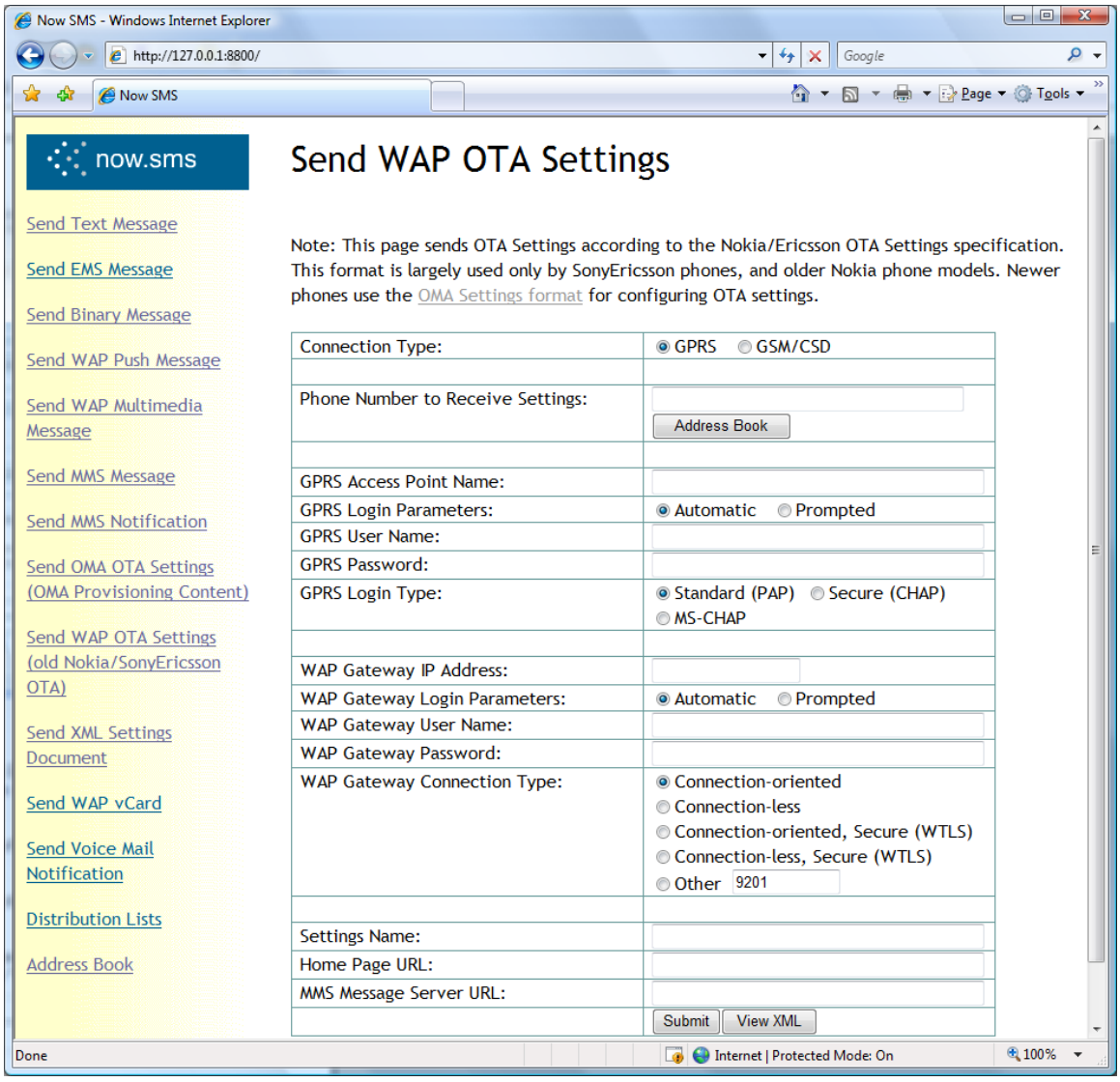
Additional information regarding WAP OTA settings can be found in **Sending WAP OTA Settings** on page 220.

Note that beginning with NowSMS 2007, there is now a "View XML" button on all of the WAP and OMA OTA Settings forms. This button will display the XML document that NowSMS has generated for the current web form. This allows for easier customisation of settings to meet different requirements.

The OTA web forms supported by NowSMS are displayed on the following pages.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

WAP OTA: Browser/MMS Client - GPRS/EDGE/Packet Data Settings



"**Connection Type**" specifies whether to use GPRS/EDGE/Packet Data, or GSM/CSD (Circuit-Switched Data). The web form is different depending on this selection, here we will describe the settings relevant for GPRS/EDGE/Packet Data.

"**Phone Number to Receive Settings**" is a comma delimited list of one or more phone numbers to receive the settings via SMS.

"**GPRS Access Point Name**" is the Access Point Name (APN) that should be used for these connection settings.

If "**GPRS Login Parameters**" is set to "Prompted", then the browser will prompt for a user name and password every time a connection is attempt, and the "GPRS User Name" and

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"GPRS Password" settings will be ignored. If it is set to "Automatic", the user will not be prompted for login information, and the "GPRS User Name" and "GPRS Password" fields will be used automatically. *(Note that many access points do not require a user name and password to be specified.)*

"GPRS Login Type" specifies the authentication protocol to be used for sending the user name and password for connecting to the access point. The standard choice is "PAP", however "CHAP" and "MS-CHAP" can also be selected.

"WAP Gateway IP Address" specifies the IP address of a WAP gateway to be used for this connection profile.

If **"WAP Gateway Login Parameters"** is set to "Prompted", then the browser will prompt for a user name and password every time a connection is attempt, and the "WAP Gateway User Name" and "WAP Gateway Password" settings will be ignored. If it is set to "Automatic", the user will not be prompted for login information, and the "WAP Gateway User Name" and "WAP Gateway Password" fields will be used automatically. *(Note that most gateways do not require a user name and password to be specified, and many phones also do not support sending a user name and password to the WAP Gateway.)*

"WAP Gateway Connection Type" specifies which of the WAP protocols to use when connecting to the WAP Gateway. For most mobile operator gateways, you would use "Connection-oriented" for a standard non-secure connection *(port 9201)*, or "Connection-oriented, Secure (WTLS)" for a secure connection *(port 9203)*. The "Connection-less" protocols provide limited functionality. For a WAP2/HTTP Proxy, the port number is installation dependent. Frequently port 8080 is used, but it can be any port value.

"Settings Name" provides a descriptive name for the connection settings, which may be displayed on the mobile phone.

"Home Page URL" specifies the home page to be configured for the settings, if these settings are to be used to configure WAP/internet browser settings.

"MMS Message Server URL" specifies the URL for the MMSC server, if these settings are to be used to configure the MMS client.

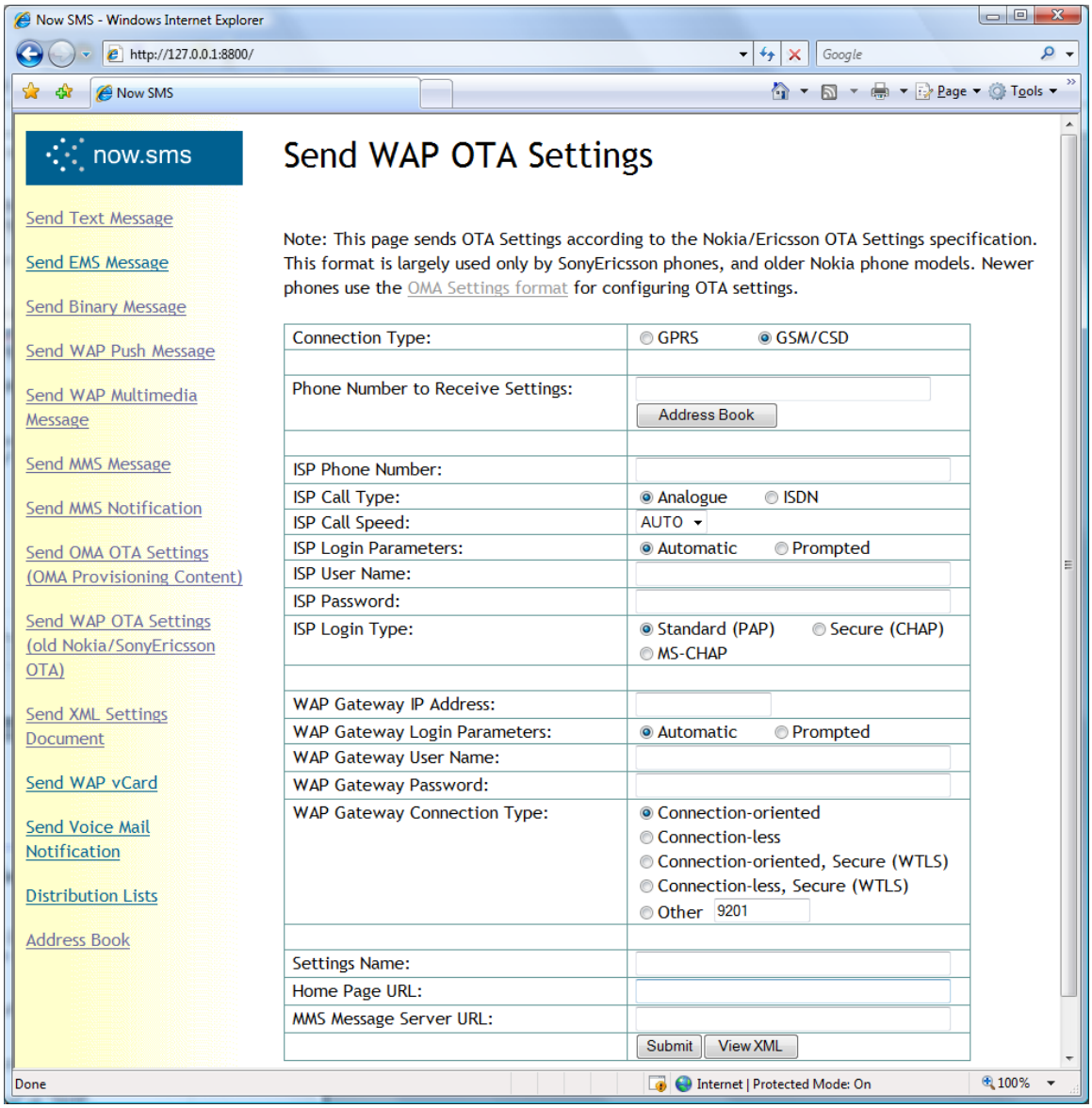
Note that some phones support both the "Home Page URL" and "MMS Message Server URL" elements being present, and will configure both the WAP/internet browser and the MMS client. Other phones will only support one of these settings being present.

Use the **"Submit"** button to send the settings, or use the **"View XML"** button to view the XML settings that are generated by this web form.

Note: To define connection settings that do not use a proxy, leave the "WAP Gateway IP Address" field blank, and set the "WAP Gateway Connection Type" to port "80".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

WAP OTA - Browser/MMS Client: GSM/Circuit Switch Data Settings



"**Connection Type**" specifies whether to use GPRS/EDGE/Packet Data, or GSM/CSD (Circuit-Switched Data). The web form is different depending on this selection, here we will describe the settings relevant for GSM/CSD.

"**Phone Number to Receive Settings**" is a comma delimited list of one or more phone numbers to receive the settings via SMS.

"**ISP Phone Number**" is the phone number to be dialed to connect to the access server.

"**ISP Call Type**" specifies whether the connection type is "Analogue" or "ISDN".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"ISP Call Speed" specifies the speed of the connection. Normally this would be set to "AUTO", but it may be necessary to configure a specific speed in special situations.

If **"ISP Login Parameters"** is set to "Prompted", then the browser will prompt for a user name and password every time a connection is attempt, and the "ISP User Name" and "ISP Password" settings will be ignored. If it is set to "Automatic", the user will not be prompted for login information, and the "ISP User Name" and "ISP Password" fields will be used automatically.

"ISP Login Type" specifies the authentication protocol to be used for sending the user name and password for connecting to the access server. The standard choice is "PAP", however "CHAP" and "MS-CHAP" can also be selected.

"WAP Gateway IP Address" specifies the IP address of a WAP gateway to be used for this connection profile.

If **"WAP Gateway Login Parameters"** is set to "Prompted", then the browser will prompt for a user name and password every time a connection is attempt, and the "WAP Gateway User Name" and "WAP Gateway Password" settings will be ignored. If it is set to "Automatic", the user will not be prompted for login information, and the "WAP Gateway User Name" and "WAP Gateway Password" fields will be used automatically. *(Note that most gateways do not require a user name and password to be specified, and many phones also do not support sending a user name and password to the WAP Gateway.)*

"WAP Gateway Connection Type" specifies which of the WAP protocols to use when connecting to the WAP Gateway. For most mobile operator gateways, you would use "Connection-oriented" for a standard non-secure connection *(port 9201)*, or "Connection-oriented, Secure (WTLS)" for a secure connection *(port 9203)*. The "Connection-less" protocols provide limited functionality. For a WAP2/HTTP Proxy, the port number is installation dependent. Frequently port 8080 is used, but it can be any port value.

"Settings Name" provides a descriptive name for the connection settings, which may be displayed on the mobile phone.

"Home Page URL" specifies the home page to be configured for the settings, if these settings are to be used to configure WAP/internet browser settings.

"MMS Message Server URL" specifies the URL for the MMSC server, if these settings are to be used to configure the MMS client.

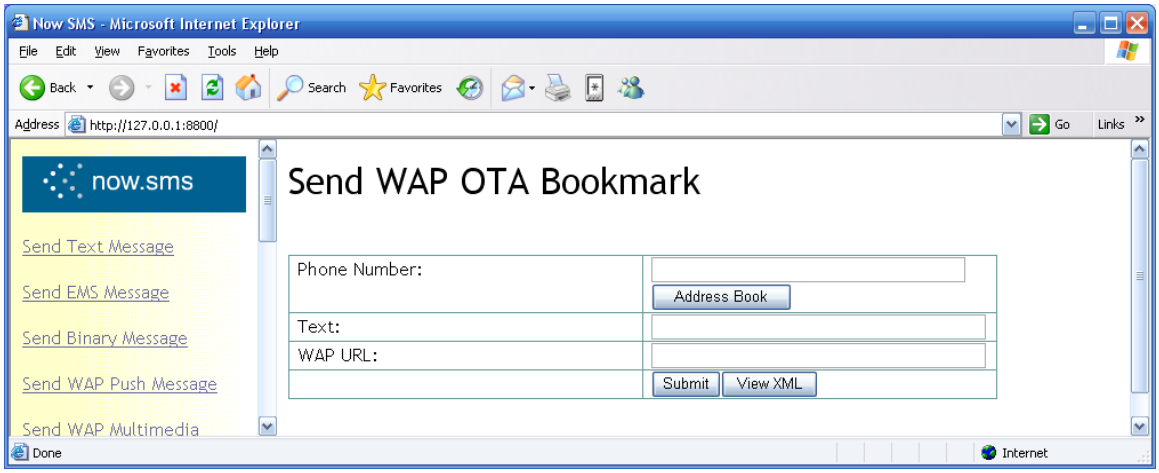
Note that some phones support both the "Home Page URL" and "MMS Message Server URL" elements being present, and will configure both the WAP/internet browser and the MMS client. Other phones will only support one of these settings being present.

Use the **"Submit"** button to send the settings, or use the **"View XML"** button to view the XML settings that are generated by this web form.

Note: To define connection settings that do not use a proxy, leave the "WAP Gateway IP Address" field blank, and set the "WAP Gateway Connection Type" to port "80".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

WAP OTA: Bookmark



"**Phone Number**" specifies a comma delimited list of one or more phone numbers to receive the bookmark.

"**Text**" specifies the descriptive name for the bookmark.

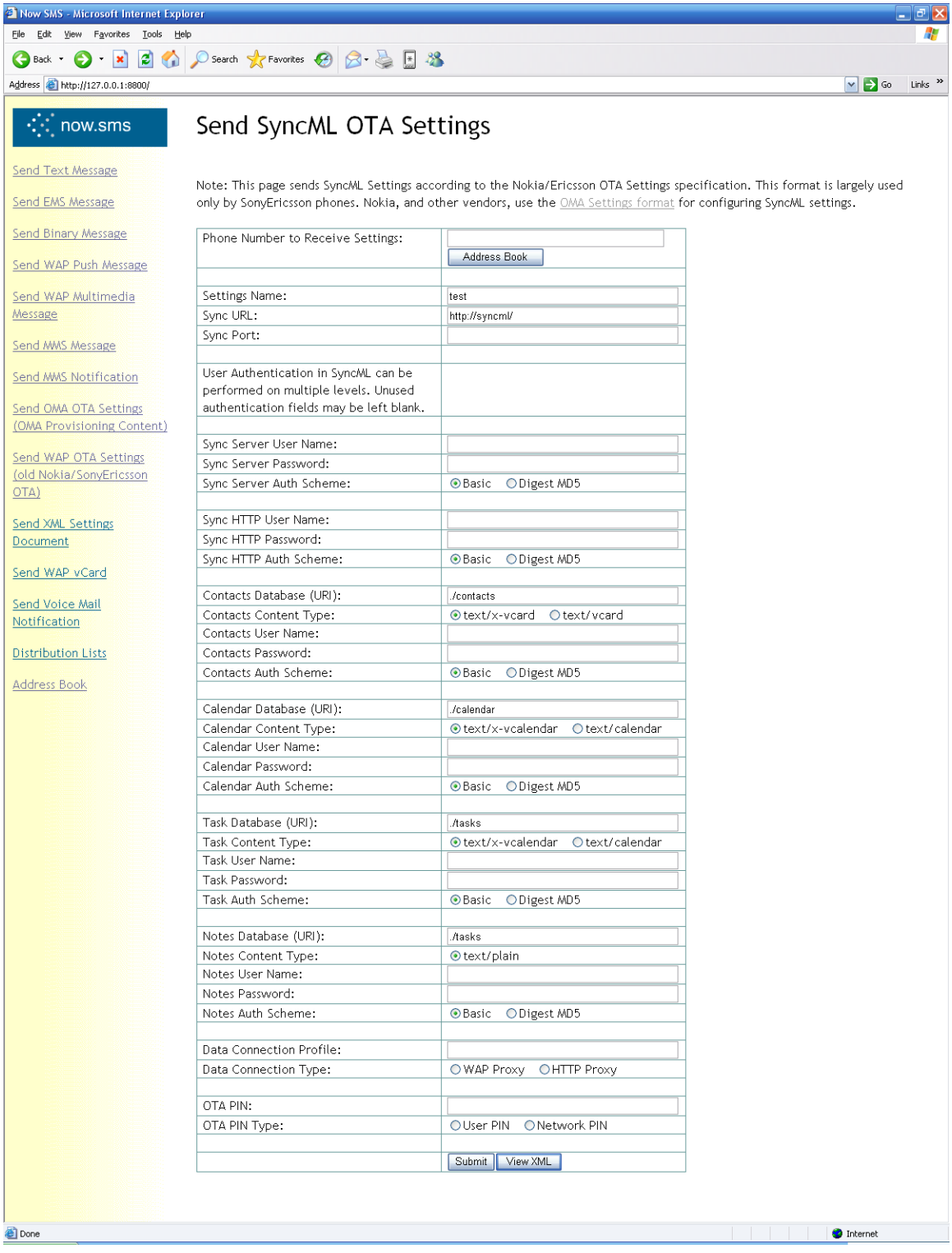
"**WAP URL**" specifies the HTTP URL for the bookmark.

Use the "**Submit**" button to send the settings, or use the "**View XML**" button to view the XML settings that are generated by this web form.

Please note that this functionality is not very widely supported, and is mostly supported only by older Nokia phones.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

WAP OTA: SyncML Settings



"**Phone Number to Receive Settings**" is a comma delimited list of one or more phone numbers to receive the settings via SMS.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"Settings Name" is a descriptive name for the settings that may be displayed on the mobile phone.

"Sync URL" is the HTTP URL associated with the SyncML server.

"Sync Port" is the port number associated with the SyncML server. Normally this would be part of the "Sync URL", but there is some ambiguity in this OTA specification.

User authentication in SyncML can be done on different levels. Therefore this OTA form contains many forms that allow a user name and password to be input. Normally, user authentication will only be performed by the SyncML Server, in which case it is only necessary to use the **"Sync Server User Name"**, **"Sync Server Password"** and **"Sync Server Auth Scheme"** settings for specifying a user name and password.

If HTTP Authentication is needed *(Authorization: header)*, then the **"Sync HTTP User Name"**, **"Sync HTTP Password"** and **"Sync HTTP Auth Scheme"** settings can be used.

A **"Database URI"** is required for each SyncML database that is supported by the server. This is a relative URI, and not a complete URL specification. (It is common to see values such as "Calendar" or "./Calendar" for the "Calendar Database URI", for example.)

The **"Data Connection Profile"** setting refers to an existing data connection profile that should exist on the device *(e.g., the "Settings Name" associated with OTA settings for a browser connection settings message that was previously sent to the device)*. The **"Data Connection Type"** specifies whether the "Data Connection Profile" uses a "WAP Proxy" or an "HTTP Proxy" (although it is unclear why this might be necessary to know in the SyncML settings).

"OTA PIN" and "OTA PIN Type" are optional parameters supported for SyncML settings over WAP OTA, but that are commonly used in OMA OTA requests.

An **"OTA PIN"** can be associated with an OTA settings message to provide a layer of authentication to the message. Many devices will allow you to send OTA settings without a PIN, but some will require a PIN to be present before the settings will be accepted.

There are three different types of OTA PINs, depending on the **"OTA PIN Type"** setting.

1.) The simplest "OTA PIN Type" is "User PIN" (USERPIN). This setting indicates that a short PIN code (often 4 digits) is supplied as the "OTA PIN". When the user receives the OTA settings message, they will need to supply this PIN code in order to be able to open the message and apply the settings.
2.) "Network PIN" (NETWPIN) indicates the PIN is a network PIN code. In the GSM environment, this is the IMSI number associated with the SIM card in the device. (Hint, if you want to experiment with determining the PIN card associated with a SIM, you can put the SIM into a GSM modem and the AT+CIMI command to return the IMSI. However, not all GSM modems support the AT+CIMI command.) When the device receives the settings, if the NETWPIN does not match the IMSI, the settings will be discarded.
3.) An additional type of PIN, known as "USERNETWPIN" also exists, which indicates a combination of the USERPIN and NETWPIN types. To use this OTA PIN type, select "Network PIN", and define the OTA PIN as the IMSI number associated with the SIM card in the device, followed by a ":" character, followed by a USERPIN *(e.g., 1234567889012345:1234)*. When the device receives the settings, the user will be

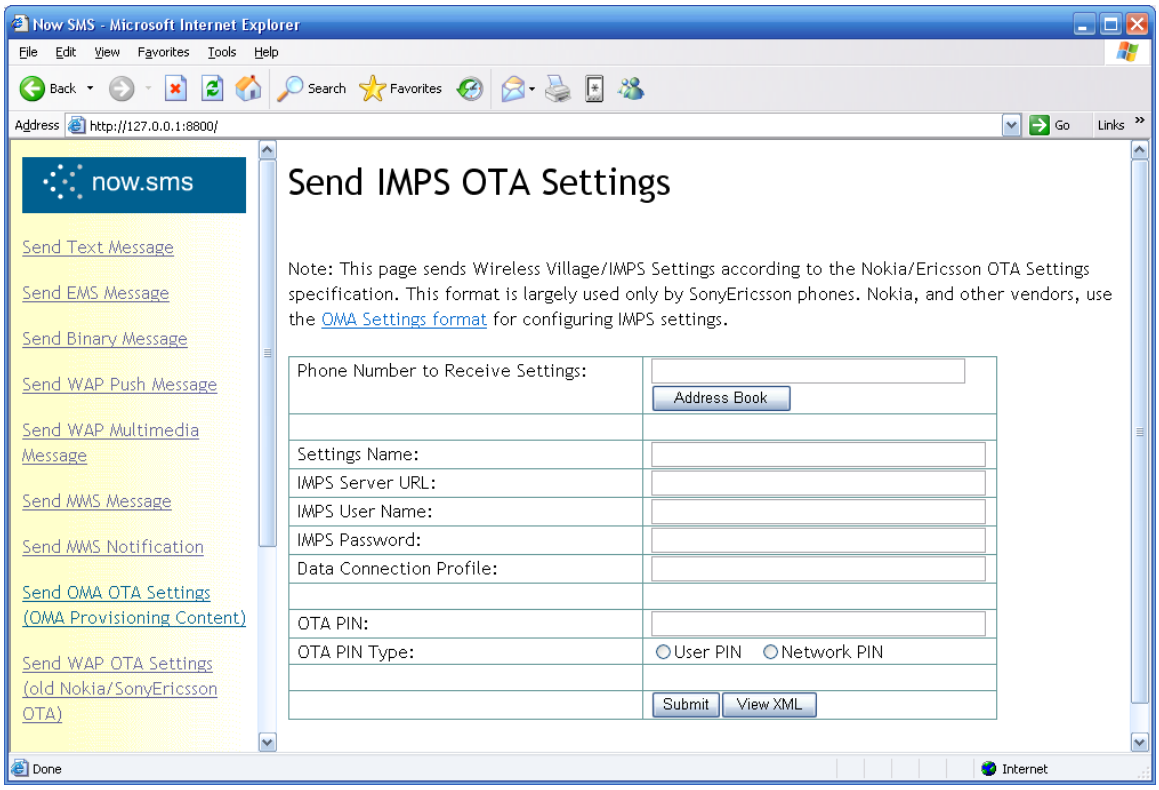
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

prompted for a PIN. This user supplied PIN, and the SIM card IMSI, must match in order for the settings to be accepted.

Use the "**Submit**" button to send the settings, or use the "**View XML**" button to view the XML settings that are generated by this web form.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

WAP OTA: Wireless Village/ IMPS Settings



"Phone Number to Receive Settings" is a comma delimited list of one or more phone numbers to receive the settings via SMS.

"Settings Name" is a descriptive name for the settings that may be displayed on the mobile phone.

"IMPS Server URL" is the HTTP URL associated with the Wireless Village/IMPS server.

"IMPS User Name" and **"IMPS Password"** define the user name and password for logging into the Wireless Village/IMPS Server.

The **"Data Connection Profile"** setting refers to an existing data connection profile that should exist on the device *(e.g., the "Settings Name" associated with OTA settings for a browser connection settings message that was previously sent to the device)*.

"OTA PIN" and "OTA PIN Type" are optional parameters supported for IMPS settings over WAP OTA, but that are commonly used in OMA OTA requests.

An **"OTA PIN"** can be associated with an OTA settings message to provide a layer of authentication to the message. Many devices will allow you to send OTA settings without a PIN, but some will require a PIN to be present before the settings will be accepted.

There are three different types of OTA PINs, depending on the **"OTA PIN Type"** setting.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

1.) The simplest "OTA PIN Type" is "User PIN" (USERPIN). This setting indicates that a short PIN code (often 4 digits) is supplied as the "OTA PIN". When the user receives the OTA settings message, they will need to supply this PIN code in order to be able to open the message and apply the settings.
2.) "Network PIN" (NETWPIN) indicates the PIN is a network PIN code. In the GSM environment, this is the IMSI number associated with the SIM card in the device. (Hint, if you want to experiment with determining the PIN card associated with a SIM, you can put the SIM into a GSM modem and the AT+CIMI command to return the IMSI. However, not all GSM modems support the AT+CIMI command.) When the device receives the settings, if the NETWPIN does not match the IMSI, the settings will be discarded.
3.) An additional type of PIN, known as "USERNETWPIN" also exists, which indicates a combination of the USERPIN and NETWPIN types. To use this OTA PIN type, select "Network PIN", and define the OTA PIN as the IMSI number associated with the SIM card in the device, followed by a ":" character, followed by a USERPIN *(e.g., 1234567889012345:1234)*. When the device receives the settings, the user will be prompted for a PIN. This user supplied PIN, and the SIM card IMSI, must match in order for the settings to be accepted.

Use the "**Submit**" button to send the settings, or use the "**View XML**" button to view the XML settings that are generated by this web form.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send OMA OTA Settings

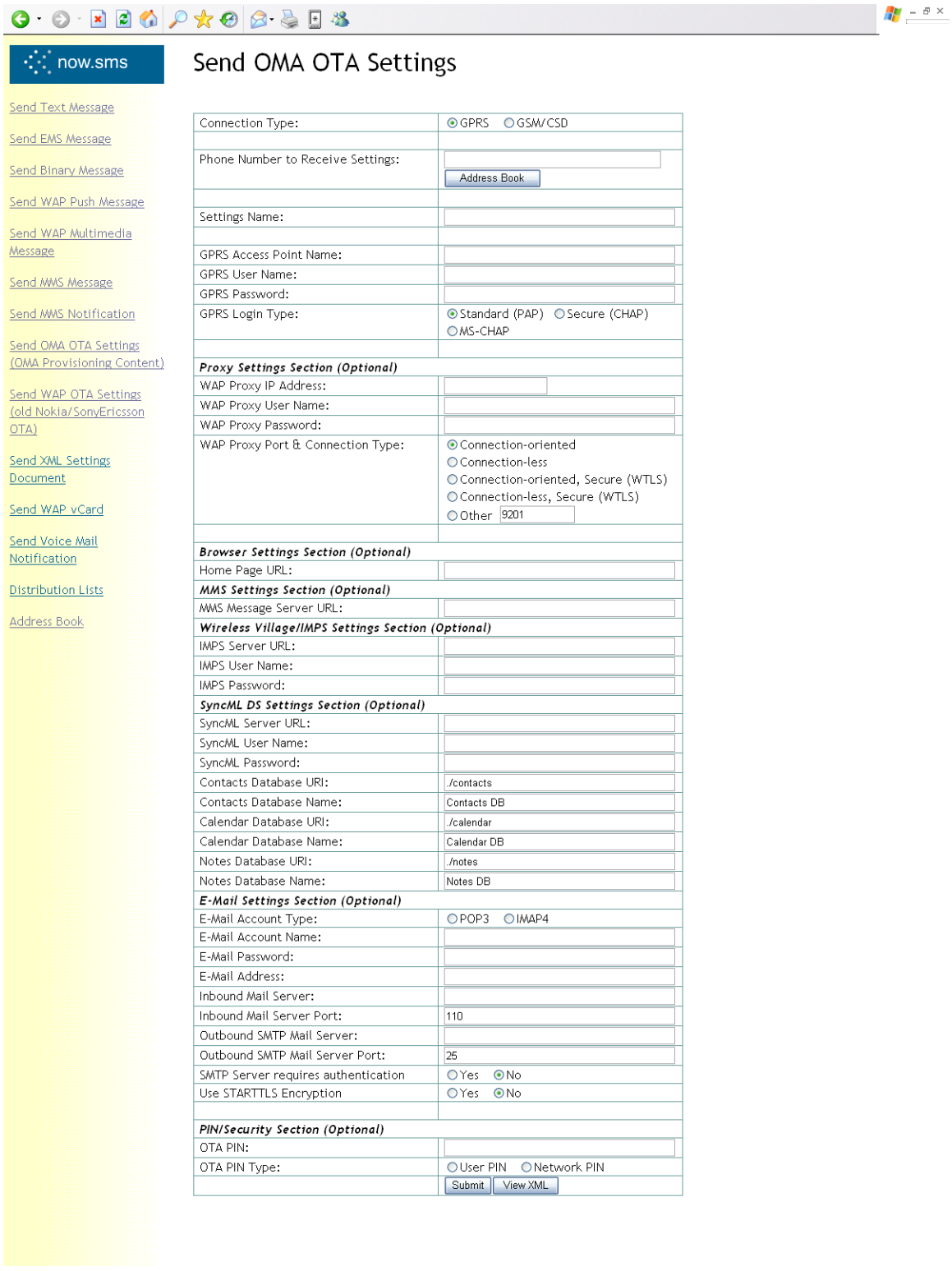
NowSMS supports sending OTA (Over-the-Air) configuration information to mobile phones that are compatible with the Open Mobile Alliance (OMA) Provisioning Content v1.1 Specification.

The "Send OMA OTA Settings" web form allows a variety of different types of configuration settings to be sent via a simple web form. Configuration settings type supported by this web form include:

- WAP/Internet Browser
- MMS Client
- Wireless Village/IMPS/Instant Messaging Client
- SyncML
- E-Mail Settings

The "Send XML Settings" web form *(page 118)* provides greater flexibility for advanced requirements, allowing any OMA Provisioning Content document to be binary encoding and sent by the gateway.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



Note that beginning with NowSMS 2007, there is a "View XML" button on all of the WAP and OMA OTA Settings forms. This button will display the XML document that NowSMS has generated for the current web form. This allows for easier customisation of settings to meet different requirements.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

It is not necessary to complete all of the sections of the "Send OMA OTA Settings" form. This form is designed to be flexible to allow you to send settings for multiple applications simultaneously, or one application at a time.

"**Connection Type**" specifies whether to use GPRS/EDGE/Packet Data, or GSM/CSD (Circuit-Switched Data). The web form is different depending on this selection, here we will describe the settings relevant for GPRS/EDGE/Packet Data.

"**Phone Number to Receive Settings**" is a comma delimited list of one or more phone numbers to receive the settings via SMS.

"**Settings Name**" provides a descriptive name for the connection settings, which may be displayed on the mobile phone.

Access Point Settings Section

The "Access Point Settings" define how the device should make its IP network connection on the packet data network. This section is normally required for each settings document. However, there is one exception. If the "Access Point Settings" are left blank, then this indicates that the settings being sent should use the "default internet" connection" that is already defined on the device.

The "default internet connection" is a connection that has a "GPRS Access Point Name" defined, but does not have any WAP or HTTP proxy associated with the connection.

If you would like to configure a "default internet connection" on a device via OMA settings, send this form with only the "Access Point Settings" section completed, and all other sections blank *(except PIN/Security Settings, if required)*.

If you are sending settings that will use a specific connection, or you cannot guarantee that a "default internet connection" is already defined in the device, you should complete this section.

"**GPRS Access Point Name**" is the Access Point Name (APN) that should be used for these connection settings.

"**GPRS User Name**" and "**GPRS Password**" specify a user name and password to be used when connecting to the APN. *(Note that many access points do not require a user name and password to be specified.)*

"**GPRS Login Type**" specifies the authentication protocol to be used for sending the user name and password for connecting to the access point. The standard choice is "PAP", however "CHAP" and "MS-CHAP" can also be selected.

Proxy Settings Section

The "Proxy Settings Section" is used if the connection sections require the use of a proxy server (WAP/WSP or WAP2/HTTP). For the purposes of this web form, "WAP Proxy" refers to either a "WAP/WSP Gateway" or a "WAP2/HTTP Proxy".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"WAP Proxy IP Address" specifies the IP address of a WAP Proxy to be used for this connection profile.

"WAP Proxy Port & Connection Type" specifies which of the protocols to use when connecting to the WAP Proxy, as well as the port number to be used. When connecting to a WAP/WSP Proxy (Gateway), you would use "Connection-oriented" for a standard non-secure connection *(port 9201)*, or "Connection-oriented, Secure (WTLS)" for a secure connection *(port 9203)*. For a WAP2/HTTP Proxy, the port number is installation dependent. Frequently port 8080 is used, but it can be any port value.

Note: Sending connection settings that do not include a proxy can require separate settings that are dependent on the receiving phone model. For most phones, leaving the "Proxy Settings Section" of the OMA OTA web form blank will allow you to define connection settings without a proxy. However, SonyEricsson and Motorola do not support the TO-NAPID parameter for browser or MMS settings. To send connection settings without a proxy to these phones, type **(blank)** into the "WAP Proxy IP Address" field *(yes, the actual word blank surrounded by parentheses),* which will enable a work-around for these phones. For Motorola phones, you must set the "WAP Proxy Port & Connection Type" to "Other" and "8080" (other values might also work) ... however for SonyEricsson phones, this setting should be left at its default value.

Browser Settings Section

This section should be completed if the settings are to configure the WAP/internet browser in the device.

"Home Page URL" specifies the home page (HTTP URL) to be associated with these settings.

MMS Settings Section

This section should be completed if the settings are to configure the MMS Client in the device.

"MMS Message Server URL" specifies the HTTP URL to be used to connect to the MMSC server.

Wireless Village/IMPS Section

This section should be completed if the settings are to configure the Wireless Village/Instant Messaging and Presence Service Client in the device.

"IMPS Server URL" is the HTTP URL associated with the Wireless Village/IMPS server.

"IMPS User Name" and **"IMPS Password"** define the user name and password for logging into the Wireless Village/IMPS Server.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

SyncML DS Settings Section

This section should be completed if the settings are to configure the SyncML Data Synchronization Client in the device.

"**SyncML Server URL**" is the HTTP URL associated with the SyncML server.

"**SyncML User Name**" and "**SyncML Password**" define the user name and password for logging into the SyncML Server.

Each SyncML Database has a "**Database URI**" *(a relative URL rather than a complete URL)* and "**Database Name**" associated with it. Leave blank if you do not wish to define one of the databases.

E-Mail Settings Section

This section should be completed if the settings are to configure the E-Mail Client in the device.

"**E-Mail Account Type**" defines whether the e-mail account uses the "POP3" or "IMAP4" protocol.

"**E-Mail Account Name**" defines the user name associated with the e-mail account on the mail server.

"**E-Mail Password**" defines the password associated with the e-mail account on the mail server.

"**E-Mail Address**" defines the e-mail address associated with the e-mail account *(e.g., user@domain.com)*.

"**Inbound Mail Server**" defines the host name or IP address of the POP3 or IMAP4 mail server from which inbound e-mail should be received.

"**Inbound Mail Server Port**" defines the port number on the inbound mail server to which the e-mail client should connect in order to receive e-mail. This port number is usually 110 for POP3 and 143 for IMAP4.

"**Inbound Mail Encryption**" specifies whether or not any encryption should be used for the POP3 or IMAP4 connection. Specify "None" for a standard non-encrypted connection. Specify "SSL/TLS" for encrypted connections using POP3 port 993 or IMAP4 port 995 (such as Gmail). Specify "STARTTLS" if your mail server uses the STARTTLS protocol to use SSL/TLS encryption over the standard unencrypted port.

"**Inbound Mail Authentication Type**" specifies the authentication protocol that is used for logging in to the mail server.

"**Outbound SMTP Mail Server**" defines the host name or IP address of the SMTP mail server to which outbound e-mail should be sent by the client.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"Outbound SMTP Mail Server Port" defines the port number on the outbound mail server to which the e-mail client should connect in order to send e-mail. This port number is usually 25.

"Outbound SMTP Encryption" specifies whether or not any encryption should be used for the SMTPconnection. Specify "None" for a standard non-encrypted connection. Specify "SSL/TLS" for encrypted connections using SMTP port 465 or 587 (such as Gmail). Specify "STARTTLS" if your mail server uses the STARTTLS protocol to use SSL/TLS encryption over the standard unencrypted port.

"Outbound SMTP Authentication Type" specifies the authentication protocol that is used for logging in to the mail server. In most cases, if the e-mail client is required to authenticate to the SMTP mail server, the "LOGIN" authentication type should be selected. Specify "None" for no authentication to the SMTP mail server, or select another protocol that is supported by your SMTP server.

PIN/Security Section

An **"OTA PIN"** can be associated with an OTA settings message to provide a layer of authentication to the message. Many devices will allow you to send OTA settings without a PIN, but some will require a PIN to be present before the settings will be accepted.

There are three different types of OTA PINs, depending on the **"OTA PIN Type"** setting.

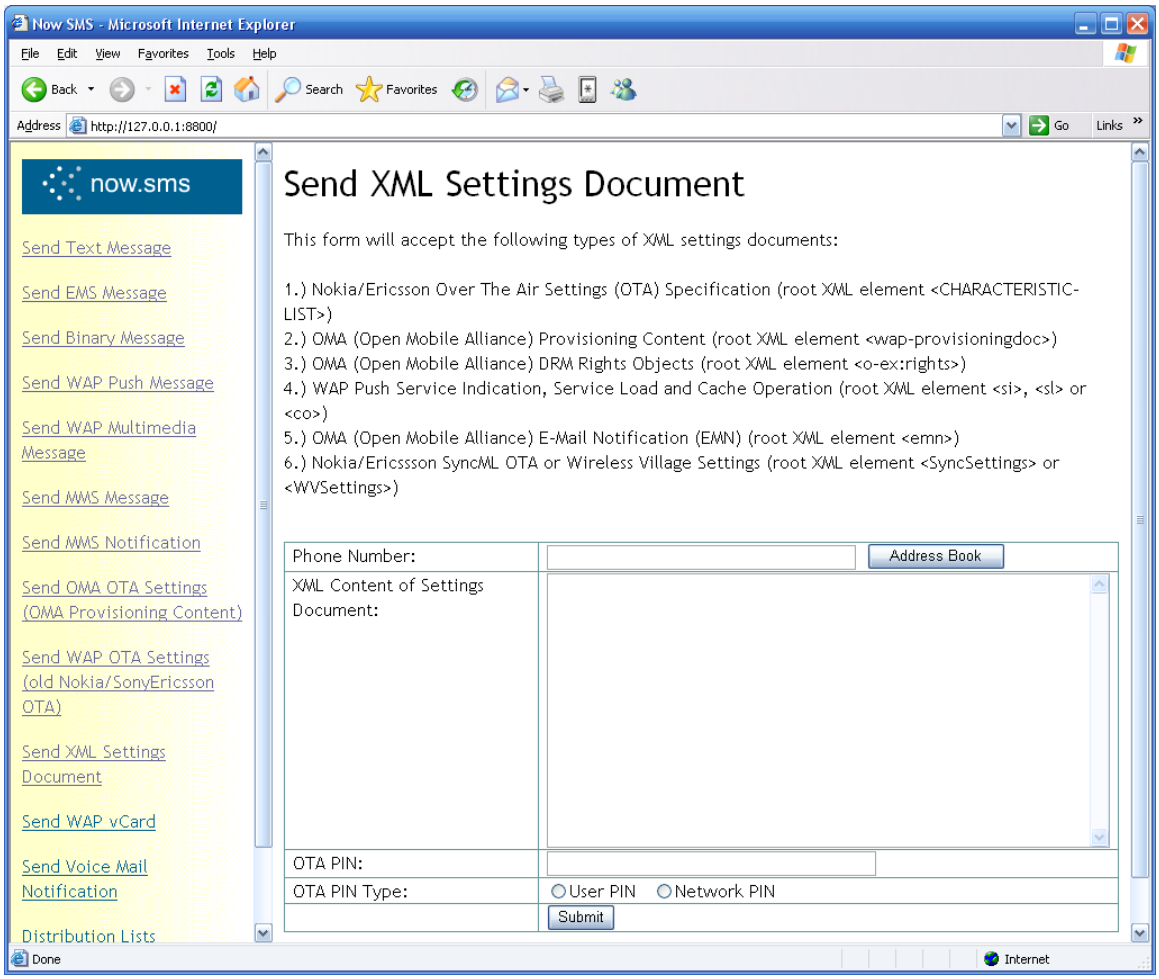
1.) The simplest "OTA PIN Type" is "User PIN" (USERPIN). This setting indicates that a short PIN code (often 4 digits) is supplied as the "OTA PIN". When the user receives the OTA settings message, they will need to supply this PIN code in order to be able to open the message and apply the settings.
2.) "Network PIN" (NETWPIN) indicates the PIN is a network PIN code. In the GSM environment, this is the IMSI number associated with the SIM card in the device. (Hint, if you want to experiment with determining the PIN card associated with a SIM, you can put the SIM into a GSM modem and the AT+CIMI command to return the IMSI. However, not all GSM modems support the AT+CIMI command.) When the device receives the settings, if the NETWPIN does not match the IMSI, the settings will be discarded.
3.) An additional type of PIN, known as "USERNETWPIN" also exists, which indicates a combination of the USERPIN and NETWPIN types. To use this OTA PIN type, select "Network PIN", and define the OTA PIN as the IMSI number associated with the SIM card in the device, followed by a ":" character, followed by a USERPIN *(e.g., 1234567889012345:1234)*. When the device receives the settings, the user will be prompted for a PIN. This user supplied PIN, and the SIM card IMSI, must match in order for the settings to be accepted.

Use the **"Submit"** button to send the settings, or use the **"View XML"** button to view the XML settings that are generated by this web form.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send XML Settings Document

The "Send XML Settings" Document provides greater flexibility for the sending out of custom XML Settings documents, above and beyond the XML that can be generated by the various OTA web forms of NowSMS.



The "Send XML Settings Document" interface supports the following types of XML settings documents:

Nokia/Ericsson Over The Air Settings (OTA) Specification up to and including v7.1

Browser and MMS settings use the root XML element <CHARACTERISTIC-LIST>
SyncML settings use the root XML element <SyncSettings>
Wireless Village/IMPS settings use the root XML element <WVSettings>

Examples of this format can be generated using the **Send WAP OTA Settings** web form, which is described in more detail beginning on page 101.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Copies of the specification that documents these settings can be downloaded from either the Nokia or SonyEricsson developer web sites.

OMA (Open Mobile Alliance) Provisioning Content

All settings use the root XML element <wap-provisioningdoc>

Examples of this format can be generated using the **Send OMA Settings** web form, which is described in more detail beginning on page 112.

The OMA Provisioning Content Specification is available for download from the Open Mobile Alliance web site at http://www.openmobilealliance.org.

OMA (Open Mobile Alliance) DRM Rights Objects

The objects use the root XML element <o-ex:rights>.

The DRMCOMP utility can be used to generate DRM rights objects. For additional information, refer to **Digital Rights Management**, beginning on page 320.

The OMA DRM Right Expression Language specification can be downloaded from the Open Mobile Alliance web site at http://www.openmobilealliance.org.

NowSMS also supports ROAP Trigger messages (root XML element <roap:roapTrigger>) as defined in the OMA DRM 2.1 specification.

WAP Push Service Indication, Service Load and Cache Operation

The operations use the root XML element <si>, <sl> or <co>.

These formats are defined in the WAP Service Indication (WAP-167), Service Load (WAP-168), and Cache Operation (WAP-175) specifications respectively.

These specifications can all be downloaded from the Open Mobile Alliance web site at http://www.openmobilealliance.org.

OMA (Open Mobile Alliance) E-Mail Notification (EMN)

This format uses the root XML element <emn>.

The OMA E-Mail Notification specification can be downloaded from the Open Mobile Alliance web site at http://www.openmobilealliance.org.

OTA PIN and OTA PIN Type

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

An "**OTA PIN**" can be associated with an OMA Provisioning Content message, in addition to some other settings message types, to provide a layer of authentication to the message. Many devices will allow you to send OTA settings without a PIN, but some will require a PIN to be present before the settings will be accepted.

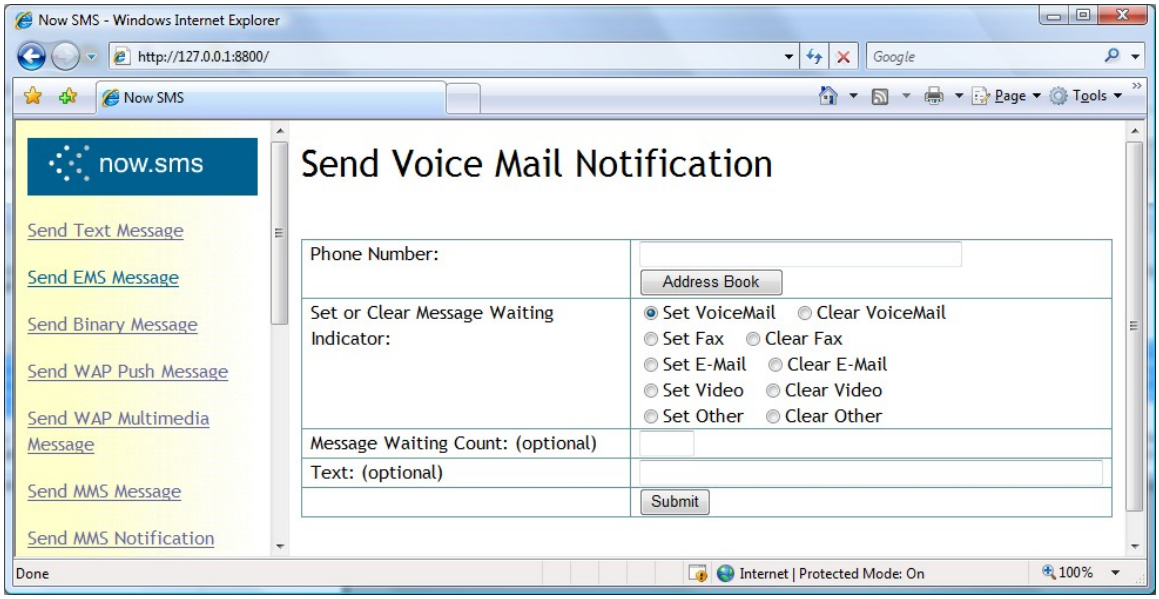
There are three different types of OTA PINs, depending on the "**OTA PIN Type**" setting.

1.) The simplest "OTA PIN Type" is "User PIN" (USERPIN). This setting indicates that a short PIN code (often 4 digits) is supplied as the "OTA PIN". When the user receives the OTA settings message, they will need to supply this PIN code in order to be able to open the message and apply the settings.
2.) "Network PIN" (NETWPIN) indicates the PIN is a network PIN code. In the GSM environment, this is the IMSI number associated with the SIM card in the device. (Hint, if you want to experiment with determining the PIN card associated with a SIM, you can put the SIM into a GSM modem and the AT+CIMI command to return the IMSI. However, not all GSM modems support the AT+CIMI command.) When the device receives the settings, if the NETWPIN does not match the IMSI, the settings will be discarded.
3.) An additional type of PIN, known as "USERNETWPIN" also exists, which indicates a combination of the USERPIN and NETWPIN types. To use this OTA PIN type, select "Network PIN", and define the OTA PIN as the IMSI number associated with the SIM card in the device, followed by a ":" character, followed by a USERPIN *(e.g., 1234567889012345:1234)*. When the device receives the settings, the user will be prompted for a PIN. This user supplied PIN, and the SIM card IMSI, must match in order for the settings to be accepted.

Use the "**Submit**" button to send the settings.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send Voice Mail Notification



Voice Mail Notification Messages are special SMS messages that are used to tell the user that they have voice mail waiting. On most mobile phones, the phone displays a message prompt, and the user can press a single key to be transferred to voice mail. This voice mail phone number is configurable via the mobile phone settings.

This web form supports sending special SMS messages to turn on and off the voice mail waiting status.

Additionally, this web form supports the ability to turn on and off other types of message waiting indicators which are commonly supported by mobile phones, including "Fax Message Waiting", "E-Mail Message Waiting" and "Video Message Waiting".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSC Messaging Server

NowSMS includes a full-featured MMSC, with support for a number of important MMS related protocols.

In order to best configure NowSMS to meet your MMS messaging needs, it is important to first have a basic understanding of how MMS works.

How MMS Works

There are two important standards that define MMS technology, one published by the 3GPP (3GPP TS 23.140), and the other a series of MMS specifications published by the Open Mobile Alliance (OMA). These two standard bodies cooperate to define the MMS protocols.

When MMS is discussed, you will often hear details of different MMS related protocols, such as MM1, MM3, MM4, MM7, as well as proprietary protocols such as EAIF, and various vendor specific proprietary variations of MM7. *(You might also wonder about MM2, MM5, MM6, MM8, MM9, and others, but they are beyond the scope of this document, and are defined in 3GPP TS 23.140.)*

MM1 is the protocol that is used between a mobile device and the MMSC Messaging Server. It defines how mobile phones send and receive messages through the MMSC.

MM3 is the protocol that is used between an MMSC and other messaging systems. It is not so much a protocol, as much as a definition of requirements for how an MMSC must be able to interoperate with other messaging systems. In the real world, this is primarily done via the SMTP e-mail protocol.

MM4 is the protocol that is used to interconnect MMSCs. It is an SMTP-based protocol with additional headers defined.

MM7 is the protocol that is used to allow Value Added Service Provider (VASP) applications to send and receive MMS messages via an MMSC. The MM7 implementation defined by the 3GPP is a SOAP-based protocol which involves the exchange of XML and MIME content over HTTP POST. Because early versions of the 3GPP MMS specifications only defined MM7 at an abstract level, several vendors of operator MMSCs have defined their own versions of MM7 which are not compatible with the MM7 SOAP-based protocol defined by the 3GPP.

EAIF is a Nokia defined proprietary protocol which extends MM1 so that it can be used by Value Added Service Providers.

As mentioned earlier, the 3GPP (http://www.3gpp.org) defines MMS in 3GPP TS 23.140. This specification defines the overall architecture of MMS. However, it defines some MMS related protocols only at a higher level architectural level, leaving out the important implementation details. In particular, the MM1 protocol is only defined in somewhat of an abstract fashion within this specification, while the MM4 and MM7 protocols are defined with complete implementation details.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The Open Mobile Alliance (http://www.openmobilealliance.org) has a series of specifications regarding MMS, and these specifications define the MM1 protocol in detail.

In practice it is the OMA specifications that provide the specific details of how mobile devices send and receive MMS messages. So to better understand MMS, it is best to first focus on this layer of communication.

MMS messages are delivered using a combination of SMS and WAP technologies.

When a mobile phone receives an MMS message, what it is actually receiving is an MMS notification message which it receives over SMS (WAP Push). This MMS notification message contains header information about the MMS message, and a URL pointer that the recipient must fetch in order to retrieve the content of the MMS message.

This URL pointer is a dynamically generated URL for the MMS message content which is stored on the MMSC. In a typical phone-to-phone MMS transaction, the process of sending and receiving the MMS message works like this:

- The sending phone initiates a data connection that provides TCP/IP network connectivity, usually over GPRS.
- The sending phone performs an HTTP POST to an MMSC of the MMS message encoding in the MMS Encapsulation Format, as defined by the Open Mobile Alliance. The encoded MMS message includes all of the content of the MMS message, as well as header information, including a list of intended recipients for the message. *(Note: In most environments, the HTTP POST will be routed through a proxy server. Some devices will use wireless profiled HTTP and TCP through a WAP 2.0 proxy server, while other devices will use the Wireless Session Protocol, WSP, through a conventional WAP proxy server/gateway.)*
- The MMSC receives the MMS message submission and validates the message sender.
- The MMSC stores the content of the MMS message and makes it available as a dynamically generated URL link.
- The MMSC generates an MMS notification message, which is sent via WAP Push over SMS to the message recipient(s). This MMS notification message contains a URL pointer to the dynamically generated MMS content.
- The recipient receives the MMS notification message. It then initiates a data connection that provides TCP/IP network connectivity (usually over GPRS).
- The recipient phone performs an HTTP (or WSP) get to retrieve the MMS message content URL from the MMSC.

You can configure NowSMS to function as an MMSC, supporting this type of user-to-user messaging traffic with the MM1 protocol. That is one type of use for the NowSMS MMSC.

However, many people look to use NowSMS as an MMSC for supporting application generated MMS messages. In these situations, NowSMS might be functioning as an MMSC, or it might be acting as a gateway for interfacing with other MMSCs. Let's explore these two types of configurations:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

1.) **Direct MMS delivery** - In this configuration, NowSMS is an MMSC. Users and/or applications submit MMS messages to the NowSMS MMSC. The MMS message content is stored on the Now SMS/MMS Gateway, and the NowSMS MMSC publishes a dynamic URL for access to the MMS message content. NowSMS generates an MMS notification message to the recipient device which is sent over SMS, and this notification includes a URL pointer back to the MMS message content on the NowSMS server.

2.) **MMS Gateway routing messages via an operator MMSC** - NowSMS supports all of the major MMS related protocols, including MM7, MM4, MM1 and EAIF for this purpose. NowSMS also supports vendor specific proprietary versions of MM7, including the non-standard variations from Ericsson, LogicaCMG, and Materna AnnyWay. NowSMS also supports a generic SMTP interface which can be used as an MM3 implementation. Any of these protocols can be used for connecting to an operator MMSC. Most frequently, at least as a starting point, what we see is the use of MM1 where NowSMS makes a GPRS connection over a GSM/GPRS modem, connects to the operator WAP gateway that is designated for MMS usage by the operator, and submits the message to the operator MMSC via the WAP gateway over the GPRS connection. *(The operator MMS gateway then generates the dynamic URL and MMS notification message that is ultimately received by the recipient device.)*

The default configuration of NowSMS is to use the first approach *(Direct MMS Delivery as an MMSC)*. When you perform direct delivery, the receiving MMS client needs to be able to connect to the NowSMS server in order to retrieve message content. In this case, it is important that the MMSC HTTP Port of the NowSMS server be accessible either over the internet or over the relevant mobile operator network(s). It is also important that the "Local Host Name or IP Address" configuration setting of the NowSMS MMSC be configured to a host name or IP address that is externally accessible.

The problem with the Direct MMS Delivery approach is that the MMS client on every mobile phone is pre-configured with settings for how the phone sends and receives MMS messages. To send or receive an MMS message, the phone makes a GPRS connection (to a GPRS APN). It then usually connects to the MMSC for sending/receiving messages through a WAP proxy/gateway. The pre-configured MMS settings on many mobile operator networks are setup to connect to a special MMS-only GPRS APN which connects to an MMS-only WAP gateway ... and this GPRS APN/WAP gateway is configured only to allow connections to the operator MMSC. If the recipient mobile phone is subscribed to an operator that has this type of setup, and you attempt direct MMS delivery, you can send the MMS notification to the phone over SMS, but the phone cannot retrieve the MMS message from your MMSC server because the GPRS APN/WAP Gateway does not allow it.

In those cases, the only alternatives are:

a.) Use the second approach, and configure NowSMS to route messages via an operator MMSC. Additional information on this type of configuration can be found in the section **Connecting to an Operator MMSC**, beginning on page 136.

b.) Change the settings in the receiving mobile phone so that it can receive messages from external MMSCs. This is usually just a matter of changing the GPRS APN and WAP Gateway IP address that is defined for the MMS client . You could change them to match the similar settings already configured for the WAP browser, which should allow access to external sites. Note that in doing this, you may no longer be able to send/receive MMS through the standard operator MMSC, so this is usually only a good solution for deployment in closed user communities.

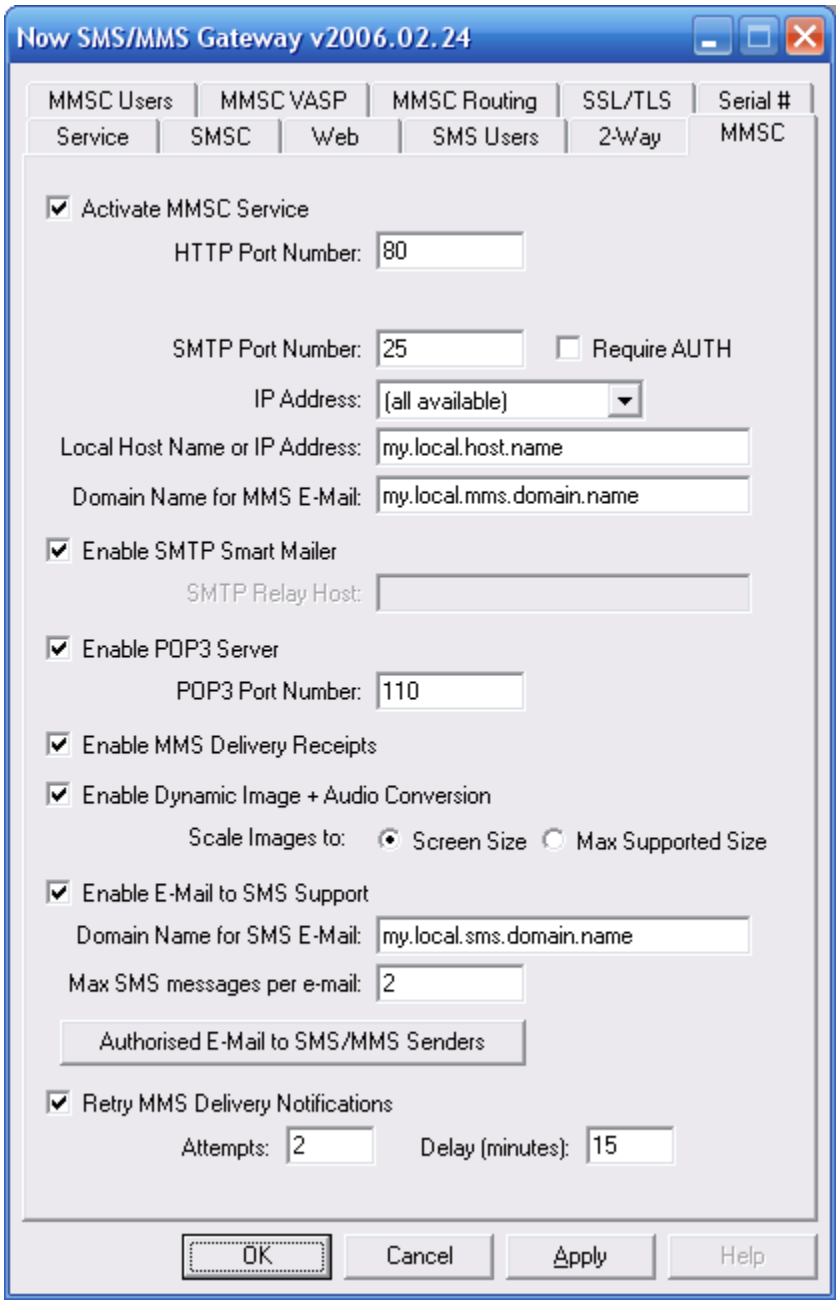
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

c.) Use an alternative to MMS, such as the Multimedia WAP push function in NowSMS. In this case, the multimedia objects are pushed to the WAP browser in the phone instead of the MMS client. NowSMS can be configured to convert a submitted MMS into this format for delivery. More information on Multimedia WAP Push can be found on page 98.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Configuring the MMSC

The "MMSC" configuration dialog specifies general configuration information for the MMSC:



The MMSC runs as a separate service process from the gateway. To activate the MMSC service, check the box next to the prompt **"Activate MMSC Service"**. *(The MMSC Service can also be activated on the "Service" page of the configuration.)*

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

When a mobile phone sends or receives an MMS message, it makes an HTTP connection to an MMSC (usually through a WAP gateway). The MMSC contains an integrated HTTP server to process these connections. Please specify an available "**HTTP Port Number**" on the local computer for the HTTP server to accept connections from mobile phones.

MMS messages can be sent to and received from, standard internet e-mail accounts. To support this functionality, the MMSC provides message format conversions between MMS and SMTP. To accept messages from internet e-mail accounts, the MMSC contains an integrated SMTP server. Please specify an available "**SMTP Port Number**" on the local computer for the SMTP server to accept e-mail messages from internet e-mail recipients. Note that the standard SMTP port number is 25, and you will require special configuration of another SMTP mail server in your network to support relaying to a port other than 25. (If you will not be using e-mail to MMS connectivity, it is still necessary to define a port number in this field. Define any available port number on the local PC.)

The PC that is running the gateway might have other web and mail services installed. For this reason, the gateway allows you to specify which of the available IP addresses on the current PC should be used by the gateway. The "**IP Address**" prompt displays the available IP addresses on the current PC. To make the gateway service available via any address on the current PC, select "(all available)", otherwise select a specific IP address.

"**Local Host Name or IP Address**" specifies the local host name or IP address of the computer that is running the MMSC service. The name or address specified here will be used to construct URLs when sending MMS messages to mobile phones. If a host name is used, this host name must be defined in DNS and resolve back to the computer running the MMSC service.

"**Domain Name for MMS E-Mail**" specifies the SMTP domain name that is associated with users defined to the MMSC. When MMS users exchange e-mail messages with internet recipients, this is the SMTP domain name associated with the MMSC users. Note that the MMSC acts as an e-mail server, and you will need to configure DNS for this domain name so that internet mail sent to this domain name is properly routed to the PC running the gateway.

When an MMS user sends a message to an SMTP recipient, the MMSC can either function as an "SMTP Smart Mailer" or it can route all outbound e-mail messages via an "SMTP Relay Host". If "**Enable SMTP Smart Mailer**" is checked, the MMSC will perform DNS lookups to locate remote recipients and perform SMTP e-mail delivery. If "Enable SMTP Smart Mailer" is not checked, it is required that an "**SMTP Relay Host**" be defined. When the smart mailer is not enabled, the MMSC will connect to the "SMTP Relay Host" to send all outbound SMTP e-mail messages. If you are using an "SMTP Relay Host", please define an appropriate SMTP mail server in your network that will perform this SMTP message relay capability.

Checking "**Enable POP3 Server**" enables the POP3 Server. The POP3 server allows user accounts defined in the "SMS Users" dialog to connect via the POP3 e-mail protocol in order to receive SMS or MMS messages *(and connect via the SMTP e-mail protocol to send SMS or MMS messages)*. More information on this setting can be found in the **E-Mail to SMS/MMS Connectivity** section on page 243. The standard "**POP3 Port Number**" is 110, however it is possible to use a non-standard port, if desired.

Checking "**Enable MMS Delivery Receipts**" should be checked if MMS delivery receipts should be supported by the MMSC. When an MMS message is sent with a delivery receipt

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

requested, the MMSC generates a delivery receipt back to the sender when the message is delivered. In some installations, it may be desirable to disable delivery receipt capabilities.

Checking "**Enable Dynamic Image and Audio Conversion**" enables the dynamic content adaptation and conversion services of the MMSC. The MMSC uses WAP/MMS "User Agent Profile" capabilities to determine the MIME formats that a device supports, as well as the maximum size of images supported by the device. Where required, the MMSC converts between common image formats (including, but not limited to GIF, JPG, PNG, BMP and WBMP) to deliver an image supported by the device. For images larger than the maximum size supported by the device, the MMSC will automatically scale the image to fit the device, speeding up download times. For audio formats, conversion between WAV and AMR is provided in the e-mail gateway interface. MIME types not supported by the receiving device, which cannot be supported, will be removed prior to delivery to the receiving device to prevent compatibility issues and unnecessary download delays.

The "**Enable E-Mail to SMS Support**" checkbox enables the SMTP interface to be used to send SMS messages. This setting is used primarily in conjunction with bulk e-mail delivery via SMTP Authentication, as described in the "**SMS User Accounts**" *(page 64)* section of this document. Gateway user accounts can be allowed to login via SMTP with an e-mail account to perform bulk delivery of SMS messages. If this configuration setting looks out of place, it is because this SMS gateway capability is provided through the MMSC's SMTP server. When enabled, specify a "**Domain Name for SMS E-Mail**", so that the gateway can identify by the domain name of the message recipient whether to route the message via SMS or MMS. Additionally, specify the "**Maximum number of SMS messages per e-mail**" to be used when converting from SMTP to SMS. When set to a value greater than 1, the gateway will use long (concatenated) SMS messages to support messages longer than 160 characters.

The "**Authorised E-Mail to SMS/MMS Senders**" can define a list of users and/or domains that are allowed to send messages to SMS or MMS recipients via the MMSC. For more information on this setting, please see **E-Mail to SMS/MMS Connectivity** section on page 243.

Checking "**Retry MMS Delivery Notifications**" tells the MMSC that it should periodically retry sending MMS notifications if a recipient of an MMS message has not retrieved the message from the MMSC.

As discussed in **How MMS Works**, the MMSC generates an MMS Notification message when it has an MMS message to be delivered to a recipient. This MMS Notification message is a WAP Push message that is usually sent over SMS. It is assumed that the SMSC that is delivering this notification message provides reliable delivery. However, in some instances, it may be desirable to configure the MMSC to retry sending the MMS notification message, in case of a problem at the SMSC or receiving device.

To enable these retries, check "**Retry MMS Delivery Notification**". Then specify the number of "**Attempts**" that the MMSC should make in retrying the MMS notification, as well as the "**Delay**" interval (in minutes) between attempts. Note that the MMSC will apply a progressive delay, increasing the delay with each attempt, such that the actual delay interval is the configured delay interval multiplied by the number of previous notification attempts. As an example, if "Attempts" is set to 4, and "Delay" is set to 15, the first retry attempt will occur after 15 minutes. The next attempt will occur after 30 minutes after

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

the previous attempt, then 45 minutes after the previous attempt, then 60 minutes after the previous attempt, after which no further notifications will be attempted.

It is also possible to configure the MMSC such that it if an MMS message is not retrieved from the MMSC within a configurable timeout period, the MMSC will then convert the message to SMS, sending a text SMS message to the recipient, with a URL like that can be used from either a phone or PC browser, to retrieve the content of the message.

To enable this SMS conversion, it is necessary to define an "MMSC Routing" of the type "Convert to SMS web Web Link".

Once that routing is defined, edit MMSC.INI, and add the following settings under the [MMSC] header:

UndeliverableRouteToSMS=VASPOutboundRouteName

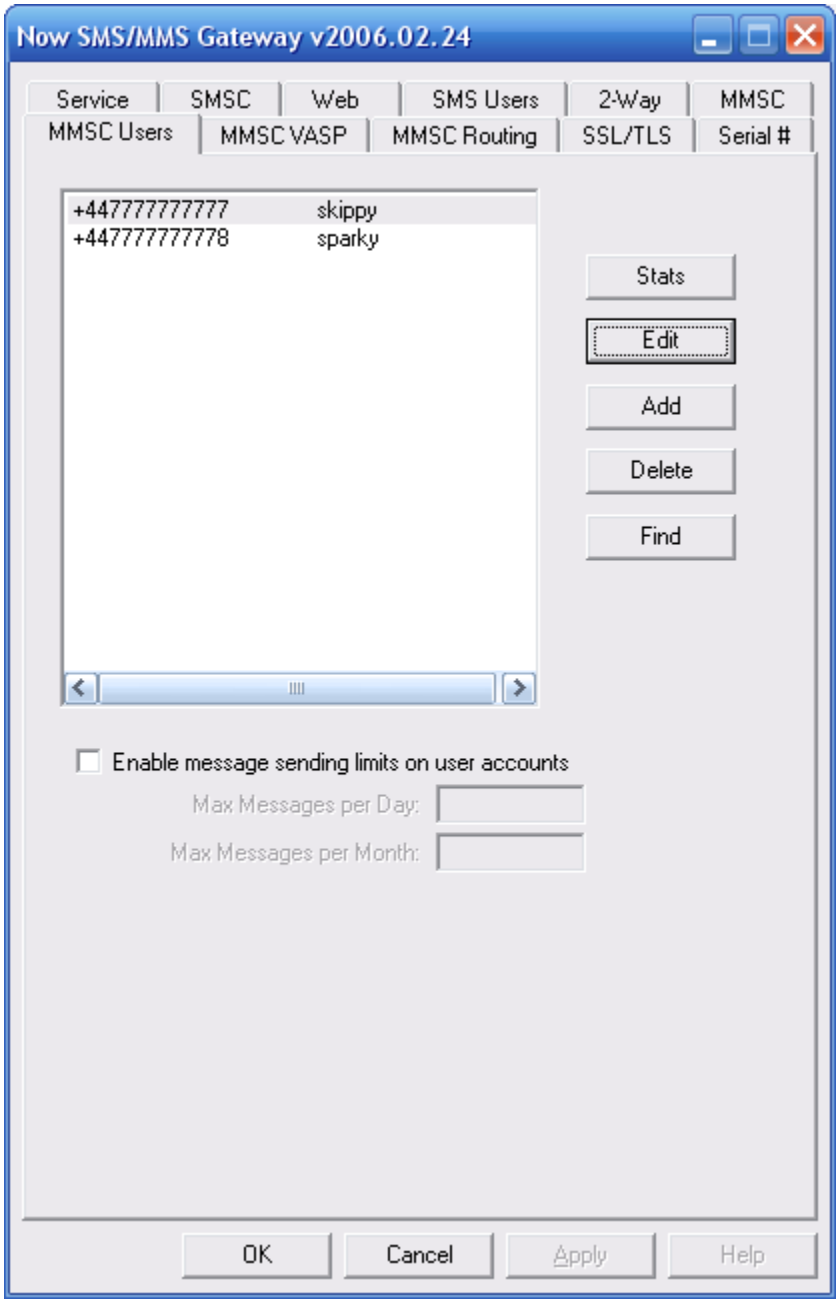
Specifies the name of an MMSC Outbound Route that is defined in the "MMSC Routing" list, which must be of the type "Convert to SMS with Web Link". By default, if an MMS message has not been retrieved within 120 minutes, the message will be rerouted to be sent as an SMS with a web link for accessing the MMS content.

UndeliverableRouteToSMSTimeout=####

is a value in minutes that changes the time period after which the UnderliverableRouteToSMS setting is applied.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The "**MMSC Users**" configuration dialog defines users that are allowed to utilize the MMSC.



To define a user to use the MMSC, you must define a phone number using international format, and an alias name for the user account. (The alias name will be used as the user name when sending and receiving SMTP e-mail.)

Note that for a mobile phone user to use the MMSC, the mobile phone user must configure their MMSC (MMS Messaging Server) to point to the address of the MMSC, and include their user name and password in the MMSC URL.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Example:

```
http://x.x.x.x:81/username=password
```

or

```
http://host.domain:81/username=password
```

The username can be either the user's alias name or phone number.

For service provider configurations, it is possible for the MMSC to read the user's phone number from the network automatically, and to automatically provision MMSC user accounts the first time a user sends an MMS message. More detail on this configuration can be found in the Technical Bulletin titled **Now MMSC Operator Considerations**, beginning on page 363.

E-Mail – MMS Gateway

The MMSC built into the gateway supports the bi-directional exchange of MMS messages between mobile phone users and internet e-mail accounts.

To enable this capability, the MMSC must be able to send and receive SMTP internet e-mail. The configuration screens for the built-in MMSC, define the required settings for sending and receiving SMTP internet e-mail.

When a mobile phone user sends a message to an e-mail recipient, the gateway will convert the message to SMTP e-mail format. Individual components of the MMS message will be sent as file attachments to the e-mail message.

To send an MMS message to a mobile phone from an e-mail client, address the message to username@mmsdomainname, where "username" is the alias name defined for the user on the "MMSC Users" dialog, and "mmsdomainname" is the "Domain Name for MMS E-Mail" defined on the "MMSC" dialog. E-mail attachments that are supported MMS content types (included in the MMSCTYPE.INI file) will be packaged and included in the MMS message sent to the mobile phone.

By default, the e-mail to MMS gateway will only accept inbound e-mail messages addressed to one of the users defined to the MMSC. It is also possible to use the SMTP interface for the bulk sending of MMS (and SMS) messages by logging into the SMTP server using an e-mail client that supports SMTP Authentication. When logged in via SMTP authentication, it is possible to send an MMS (or SMS) message to any recipient, by sending to addressing the message to "phonenumber@mms.domain.name", where "mms.domain.name" is the "Domain Name for MMS E-Mail" defined on the MMSC configuration dialog *(page 122)*. An authenticated SMTP user can send an SMS message by addressing the message to "phonenumber@sms.domain.name", where "sms.domain.name" is the "Domain Name for SMS E-Mail" defined on the MMS configuration dialog *(page 122)*. To define a user account that is allowed to login with SMTP Authentication, refer to the **"SMS Users"** configuration dialog *(page 64)*.

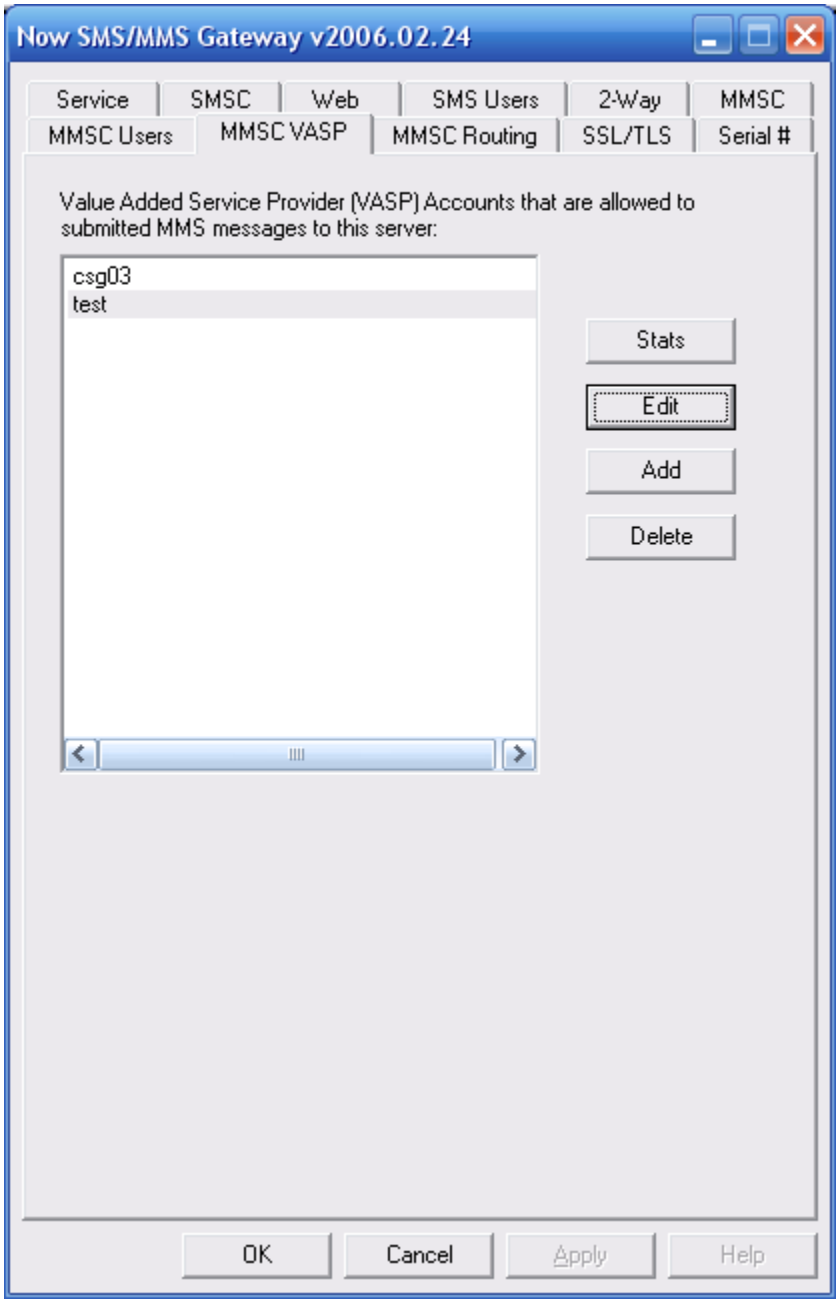
Additional information can be found in the section titled **E-Mail to SMS/MMS Connectivity**, begging on page 243.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Configuring MMS VASP Accounts

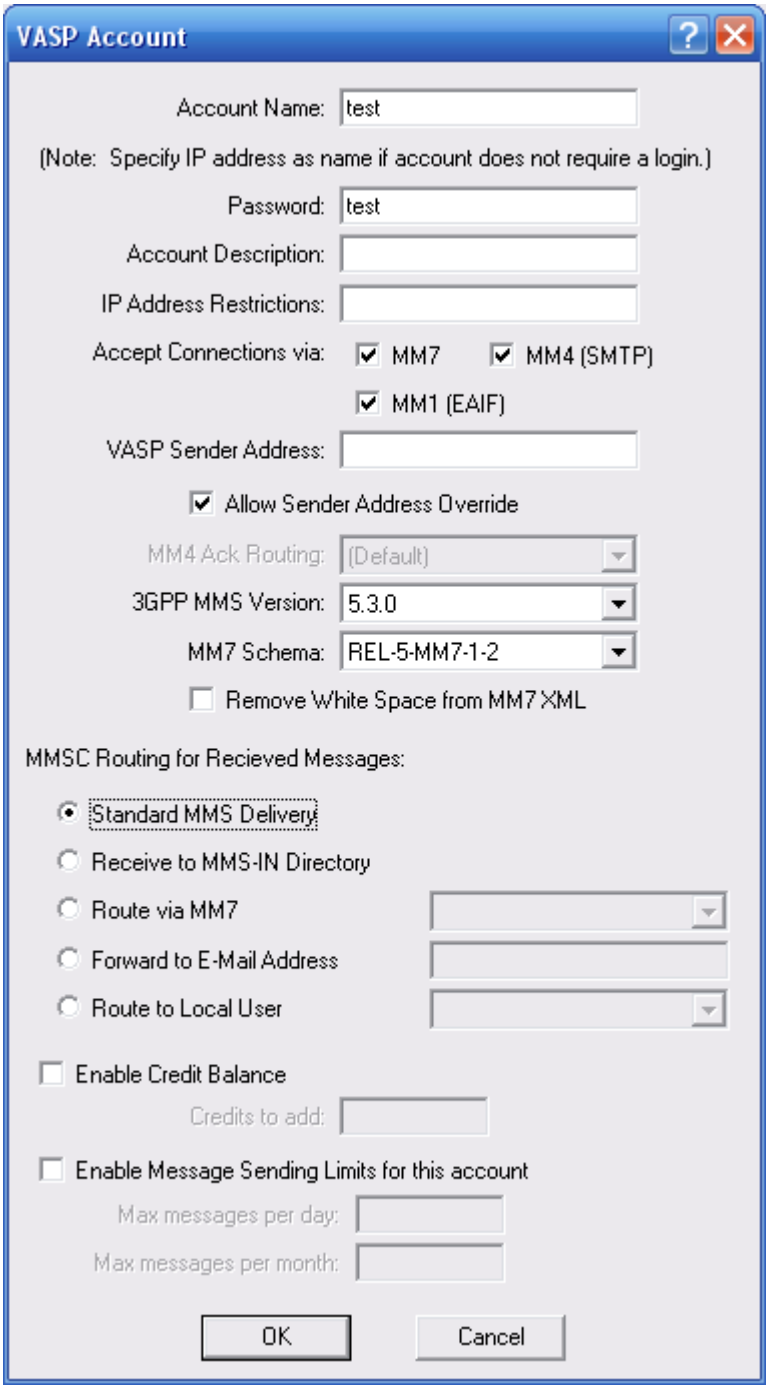
Value Added Service Providers (VASPs) are external application services that need to submit MMS messages to the Now SMS/MMS Gateway.

A VASP could be a customer that is submitting MMS messages through your gateway, or in some cases, a VASP account could be an operator MMSC delivering MMS messages to your gateway. VASP accounts are defined on the "MMSC VASP" configuration dialog.



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

When you "Add" or "Edit" a VASP connection, the following screen is displayed:



"**Account Name**" specifies a login name for the VASP to login to your Now SMS/MMS Gateway.

In some cases, where a VASP does not have configuration settings to login to the gateway, you should specify an IP address. When an IP address is specified as the "Account Name",

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

the Now SMS/MMS Gateway will treat any connections from the specified IP address as being from this defined VASP.

"**Password**" specifies a password for the VASP to login to your Now SMS/MMS Gateway.

"**Account Description**" is a descriptive field that can be associated with the VASP. It is not used for any purposes of the VASP making a connection to the gateway.

"**IP Address Restrictions**" specifies one or more IP addresses from which the account is allowed to login to the MMSC. If multiple IP addresses are specified, they should be separated by a comma only, with no white space characters. The "*" character is permitted as a wildcard to allow access from all IP addresses on a particular subnet.

VASPs can connect to the MMSC using any of the following protocols: MM7, MM4, MM1 or EAIF. The "**Accept Connections via**" setting allows you to select the protocols that this particular VASP is allowed to use when connecting to the MMSC. (Note: As EAIF and MM1 are very similar protocols, if MM1 is enabled, EAIF is automatically enabled, and vice versa.)

When a VASP connects to the MMSC using the MM7 protocol *(see page 180)*, it should make connections to the configured "HTTP Port Number" on the "MMSC" configuration dialog, using a URL path of "/mm7" (e.g., http://host.name:port/mm7). Unless an IP Address is specified as the "Account Name", the application should authenticate to the MMSC using HTTP Basic Authentication using the configured "Account Name" and "Password" for the VASP account.

When a VASP connects to the MMSC using the MM4 protocol *(see page 183)*, it should make connections to the configured "SMTP Port Number" on the "MMSC" configuration dialog. Unless an IP address is specified as the "Account Name", the application should use "SMTP AUTH" to authenticate to the MMSC.

When a VASP connects to the MMSC using the EAIF protocol *(see page 186)*, it should make connections to the configured "HTTP Port Number" on the "MMSC" configuration dialog, using a URL path of "/eaif". Unless an IP Address is specified as the "Account Name", the application should authenticate to the MMSC using either HTTP Basic Authentication, or it can include the username and password in the URL using the following format (http://host.name:port/eaif/username=password).

When a VASP connects to the MMSC using the MM1 protocol *(see page 185)*, it should make connections to the configured "HTTP Port Number" on the "MMSC" configuration dialog, using a URL path of "/mm1". Unless an IP Address is specified as the "Account Name", the application should authenticate to the MMSC using either HTTP Basic Authentication, or it can include the username and password in the URL using the following format (http://host.name:port/mm1/username=password).

"**VASP Sender Address**" specifies a default sender address for messages submitted by this VASP. If "**Allow Sender Address**" is checked, this "VASP Sender Address" is applied as the message sender address only if the VASP did not supply a sender address in a message submission. If "Allow Sender Address" is not checked, the "VASP Sender Address" is applied as the sender address for all messages submitted by the VASP, overriding any sender address specified by the VASP.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"MM4 Ack Routing" applies only to VASP accounts that are defined for receiving messages via MM4. When defining an MM4 connection between two operator MMSCs, it is normal to define both an inbound "MMSC VASP" and outbound "MMSC Routing" definition. In order to ensure that MM4 message acknowledgments are delivered correctly, this field should identify the corresponding "MMSC Routing" definition through which MM4 messages are routed back to the other MMSC.

"3GPP MMS Version" controls the MMS Version that the MMSC uses when generating MM7 responses. Set this value only if the VASP requires a specific MMS version setting.

"MM7 Schema" controls the MM7 XML Schema that is used when generating MM7 responses. Set this value only if the VASP requires a specific MM7 schema.

"Remove White Space from MM7 XML" - The MMSC normally generates MM7 XML in a user friendly format that includes line breaks. Some applications do not like any white space or line breaks within the MM7 XML, and this setting forces any white space to be removed from the XML.

When the Now SMS/MMS Gateway receives a message from a VASP account, five options are available for processing the received message.

"Standard MMS Delivery" means the message will be queued for outbound delivery by the MMSC as would any other MMS message submitted to the gateway.

"Receive to MMS-IN Directory" means that the message will be received and stored to the "MMS-IN" subdirectory of the gateway as a received message file, and will not be delivered externally to the gateway.

"Route via MM7" means that the message will be received and routed to an MM7 connection that is defined in the "MMSC Routing" configuration dialog. (Note: This can also include a 2-way PHP script, as described on page 391.)

"Forward to E-Mail Address" means that the message will be forwarded to a specified e-mail address.

"Route to Local User" converts the MMS message into an e-mail format so that it can be retrieved by a local user account ("SMS Users") connecting to the NowSMS server via POP3. For more information, please see **E-Mail to SMS/MMS Connectivity** on page 243.

It is possible to define a "Credit Balance" for each VASP by checking **"Enable Credit Balance"**. This balance specifies a fixed number of messages that the account is allowed to send. Each time the VASP sends a message to a recipient, a credit is deducted from this balance. To add or remove credits, enter the number of credits in the **"Credits to add"** field and press **"Ok"**. *(Prefix the number with a minus symbol, -, to remove credits from an account.)*

It is also possible to limit the number of messages a VASP can send in a daily or monthly period. To enable daily and monthly limits, check **"Enable Message Sending Limits for this account"**, and provide sending limit values for **"Max messages per day"** and **"Max messages per month"**.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Connecting to an Operator MMSC

The Now SMS/MMS Gateway includes an MMSC which can function independent of an operator MMSC. However, in many scenarios it is useful to configure the gateway to send and/or receive MMS messages via an operator MMSC.

Connecting to an Operator MMSC – Using a GPRS Modem

The Now SMS/MMS Gateway can use a GSM/GPRS modem to send and receive MMS messages via an operator MMSC. In this type of configuration, there is no special setup requirement required by the mobile operator. The Now SMS/MMS Gateway sends and receives MMS messages using the same protocol that is used by the MMS client in a mobile phone, so it simply requires that the SIM card in your mobile phone be provisioned by your mobile operator for MMS support.

It is also helpful to have an MMS compatible mobile phone that is already configured to send and receive MMS messages via the operator MMSC. In this way, you can check the settings on the mobile phone to determine the correct settings to configure in the Now SMS/MMS Gateway to allow the gateway to connect to your operator MMSC.

In this scenario, when sending an MMS message, the Now SMS/MMS Gateway initiates a GPRS connection to the mobile operator, it then makes a connection to the operator WAP gateway, and submits the MMS message over this WAP and GPRS connection to the mobile operator's MMSC.

When receiving an MMS message, the gateway first receives an MMS notification message via SMS. When this special notification message is received, the gateway initiates a GPRS connection to the mobile operator, and a connection to the operator WAP gateway over GPRS, in order to retrieve the content of the MMS message from the mobile operator's MMSC.

There are two areas where MMS settings specific to using a GSM/GPRS modem are specified, one is specific to sending messages, and the other is specific to receiving messages.

To send MMS messages to an operator MMSC via a GPRS modem connection, please see the description below for **Connecting to an Operator MMSC – Sending MMS Messages** *(page 142)*. This section focuses on setting for receiving MMS messages over a GPRS modem connection.

When a GSM phone or GSM modem receives an MMS message, what it actually receives is an MMS notification message. The MMS notification message arrives via an SMS message (usually two concatenated SMS messages). The MMS notification message contains header information about the MMS message, plus a URL pointer (e.g., http://host.name/path/file.mms) to the actual MMS content. In order to receive the MMS message, the receiving device must initiate an HTTP connection to retrieve the URL with the MMS message content (this URL generally points to content residing on the operator MMSC).

In earlier releases of the Now SMS/MMS Gateway, when the gateway received an MMS notification message, it would always attempt to retrieve the MMS message content via

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

HTTP over the default internet connection of the PC running the gateway software. While this worked for some installations, this did not work for retrieving MMS message content from many operator MMSCs. The reason that it would not work in many installations is that the operator MMSCs were either firewalled off so that they were inaccessible from the internet, or they were on private IP addresses within the operator network.

The v5.0 release of the Now SMS/MMS Gateway includes configuration options that allow the gateway to automatically initiate a GPRS connection when an MMS notification message is received. Rather than retrieving the MMS message via HTTP, the gateway can make a connection to the operator WAP gateway to retrieve the MMS message content from the MMSC through the WAP gateway using WAP protocols. As almost all MMS compatible phones issue their MMS requests through a WAP gateway proxy, this allows the gateway software to appear just like a standard MMS compatible phone.

Note that you can still not use an MMS compatible phone as a GSM modem if you want to be able to receive MMS messages through the gateway. This is because an MMS compatible phone intercepts the MMS notification and tries to process it automatically, never forwarding it to the gateway. You must use a GSM/GPRS modem device to take advantage of this feature.

Before continuing, you will need to determine the GPRS APN (Access Point Name), the IP address of the WAP gateway, and the MMSC Message Server URL that are used for sending/receiving MMS via your operator network. These settings are operator dependent, and it may be advisable to check the MMS configuration settings on a working mobile phone to determine the correct settings. Note that your mobile operator possibly has multiple GPRS APNs and multiple WAP gateways, and you need the settings that are appropriate for MMS, not for WAP browsing or general internet connectivity.

With most GSM/GPRS modems (except for some PC card modems that allow GPRS access through a network connection without using dial-up networking), NowSMS cannot simultaneously use the modem for both sending/receiving SMS messages and accessing GPRS data.

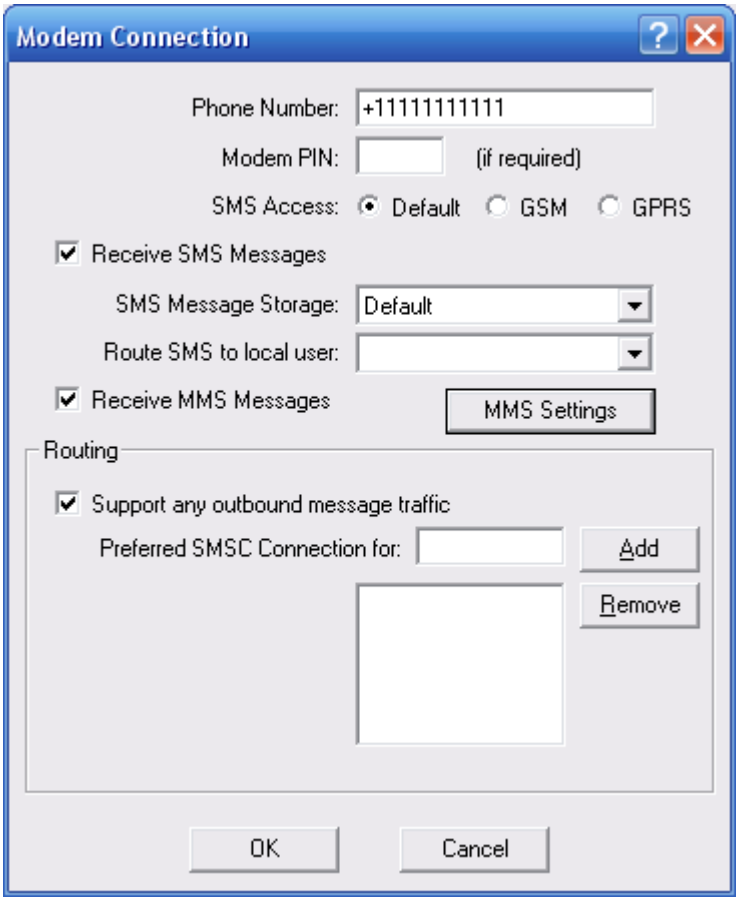
The gateway allows you to define separate settings for receiving MMS messages for each SMSC connection that is defined to your gateway. (Recall that the first stage of receiving MMS messages is that an MMS notification message is received via SMS.)

From the "**SMSC**" page of the Now SMS/MMS Gateway configuration dialog, highlight a connection for which you want to configure settings for receiving MMS messages, and select "Properties".

The screen that is displayed will vary depending on the type of SMSC connection selected. But for any of the types of SMSC connections that can receive SMS messages, there will be a check box for whether or not the gateway should attempt to "Receive MMS Messages" for MMS notifications received via that SMSC connection.

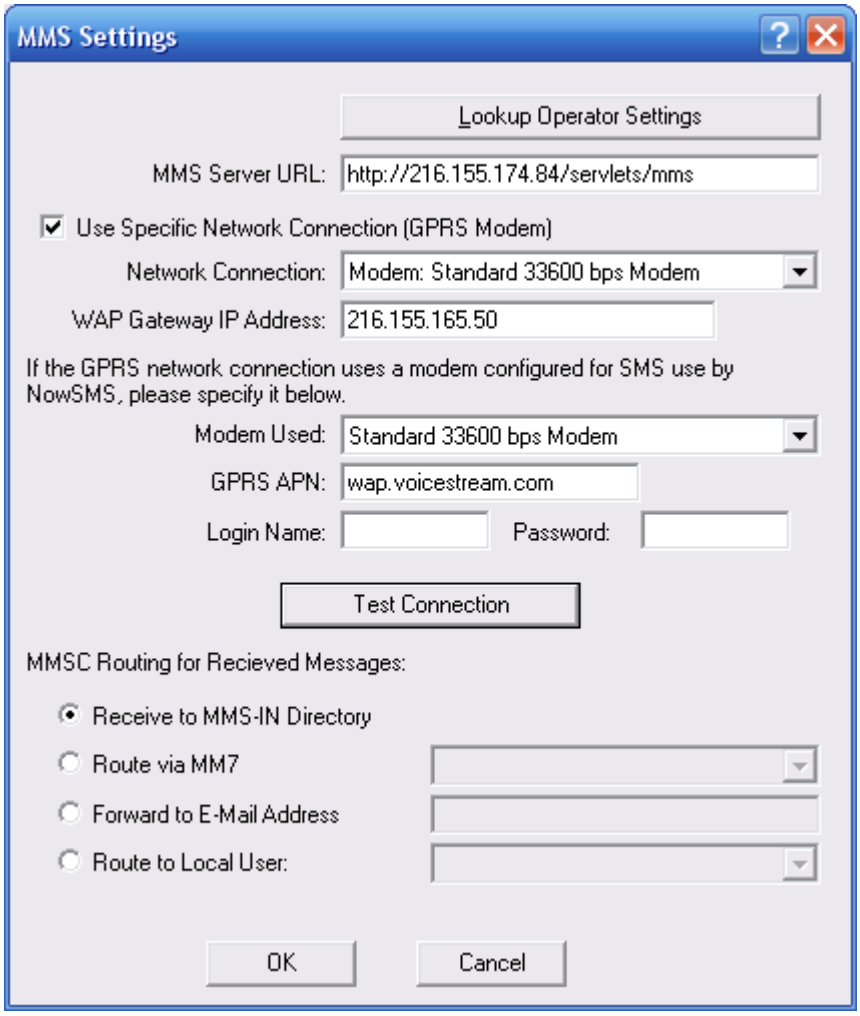
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

To configure the "MMS Settings", ensure that "Receive MMS Messages" is checked, and then select the "MMS Settings" button.



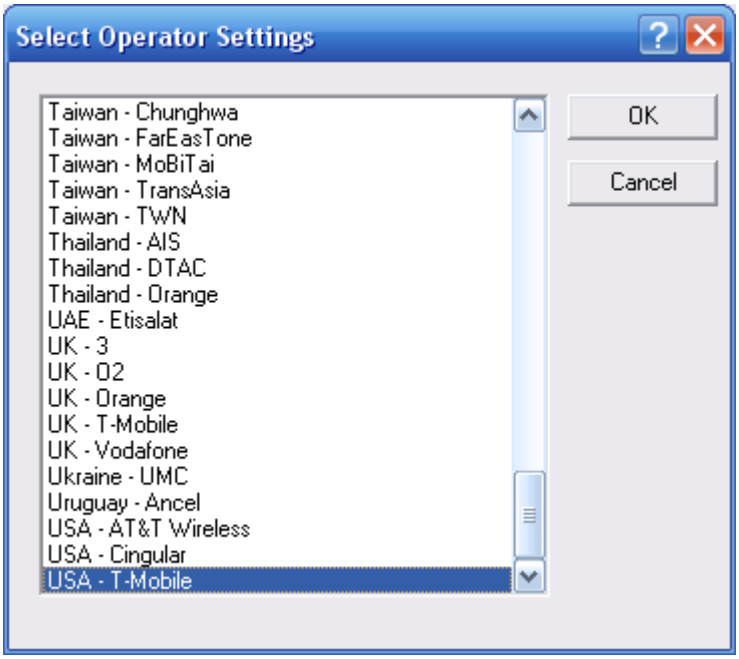
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The "**MMS Settings**" dialog specifies how to receive MMS messages associated with any MMS notifications received over the SMSC connection.



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The "**Lookup Operator Settings**" button will pre-load MMS settings for selected mobile operators.



The "**MMS Server URL**" is the URL address for the operator MMSC. While this setting is primarily used when sending MMS messages, the gateway will acknowledge MMS message receipt to this URL, and in many cases, if the operator MMSC does not receive this acknowledgment it could continue sending the same message repeatedly, or delay the sending of future MMS notifications.

If the MMS message content needs to be received over a specific network connection, such as a GPRS modem, check the "**Use Specific Network Connection (GPRS Modem)**" checkbox.

In the "**Network Connection**", select the name of the network connection that is to be used. NowSMS can use any of three different types of connections to make a GPRS connection. The different available connections are listed in the drop-down field associated with this configuration field. The different types of connections are prefixed with the text "**Modem:**", "**Dial-up:**", or "**Network:**", and are described below:

a.) "**Modem:**" refers to a standard GPRS connection to be initiated over a GPRS modem. Select the modem that should be used for this connection. (Note that only modems that have a Windows modem driver defined for the modem can be used. If your modem does not have a modem driver supplied by the manufacturer, you can use one of the "Standard" or "Generic" modem drivers available when defining a modem in the Windows Control Panel.)

b.) "**Dial-up:**" refers to a dial-up networking connection defined on the current PC. This setting can be used if you have advanced requirements and wish to create a custom dial-up networking profile.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

c.) **"Network:"** refers to a particular network interface card installed in the PC. Some PC card GPRS modems, such as the Sierra Wireless Aircard 750 provide GPRS access via a network driver interface. To tell NowSMS to use that specific network driver for connecting to the MMSC, select the named "Network:" driver.

The **"WAP Gateway IP Address"** field should contain the IP address of the operator WAP gateway which will act as a proxy for connections to the MMSC. If the gateway uses WAP2/HTTP instead of WSP, prefix the address with "http://", using the format "http://ip.address:port".

For most GSM/GPRS modems, it is not possible for the Now SMS/MMS Gateway to connect to the modem to send/receive SMS messages as the same time as a GPRS connection is active (Network Connection type = "Modem:" or "Dial-up:"). For these situations, the "Modem Used" setting should specify the name of a modem that is used by the Dial-up networking profile. If the Now SMS/MMS Gateway is using the modem to send/receive SMS messages, it will automatically release the modem when it needs to initiate a GPRS connection, otherwise the GPRS connection will not be able to be properly established.

The **"GPRS APN"** field specifies the GPRS Access Point Name (APN) to be accessed for connecting to the MMSC. This setting is operator dependent, and it may be advisable to check the MMS configuration settings on a working mobile phone to determine the correct settings. Note that your mobile operator possibly has multiple GPRS APNs and multiple WAP gateways, and you need the settings that are appropriate for MMS, not for WAP browsing or general internet connectivity. Note that this setting is only available when using a Network Connection of type "Modem:". For other connection types, the GPRS APN must be configured for the connection using some means external to NowSMS.

The **"Login Name"** and **"Password"** parameters specify a username and password to be used for connecting to the GPRS network.

The **"Test Connection"** dialog verifies that the Now SMS/MMS Gateway can initiate a network connection to the specified profile, and that it can make a connection to the specified WAP gateway over the connection. (The "MMS Message Server URL" is not tested at this time.)

When the Now SMS/MMS Gateway receives an MMS message, four options are available for processing the received message.

"Receive to MMS-IN Directory" means that the message will be received and stored to the "MMS-IN" subdirectory of the gateway as a received message file, and will not be delivered externally to the gateway.

"Route via MM7" means that the message will be received and routed to an MM7 connection that is defined in the "MMSC Routing" configuration dialog. (Note: This can also include a 2-way PHP script, as described on page 391.)

"Forward to E-Mail Address" means that the message will be forwarded to a specified e-mail address.

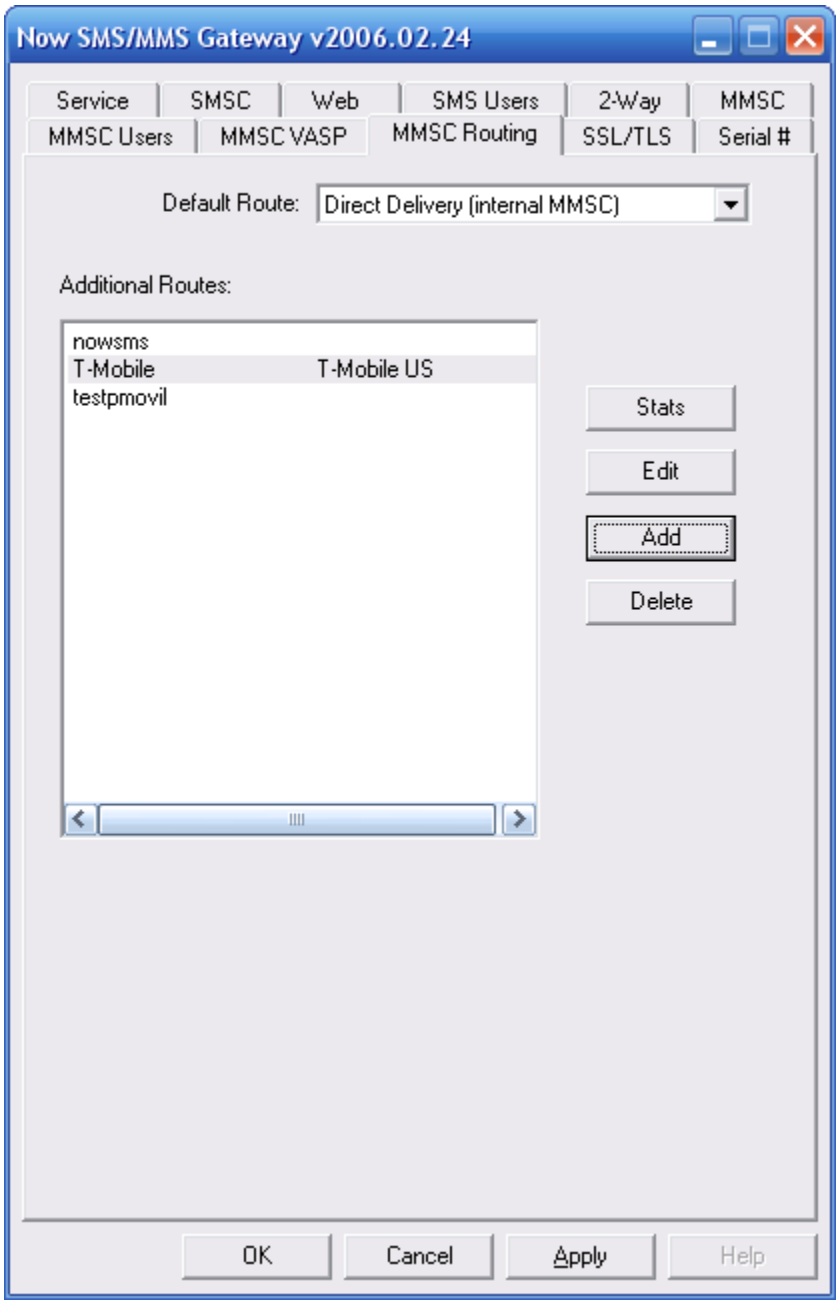
"Route to Local User" converts the MMS message into an e-mail format so that it can be retrieved by a local user account ("SMS Users") connecting to the NowSMS server via POP3. For more information, please see **E-Mail to SMS/MMS Connectivity** on page 243.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Connecting to an Operator MMSC - Sending MMS Messages

Similar to how the Now SMS/MMS Gateway allows recipient phone number masks to be defined to route SMS messages to different SMSC connections, the MMSC allows recipient phone number masks to be defined to route MMS messages to different MMSC connections.

The "**MMSC Routing**" page of the gateway configuration dialog specifies outbound routes for MMS messages.



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

By default, the Now SMS/MMS Gateway will act as an MMSC and perform direct delivery of MMS messages to recipients using a combination of SMS and WAP technologies.

However, there are certain closed operator configurations where it can be desirable to route MMS messages via an operator MMSC, or to reformat MMS message content to be delivered as a multimedia WAP push.

The "**Default Route**" setting specifies the default route to be used for delivering MMS messages. This default route is used whenever the MMSC has an MMS message to deliver, and there is no matching route to handle the message.

The MMS routing logic first checks to see if the recipient is listed in the "MMSC Users" list. If the recipient is an existing MMSC user, then the MMSC will perform direct MMS delivery to the recipient. *(Note: If DisableMMSDirectDelivery=Yes is set under the [MMSC] heading of MMSC.INI, this check of the recipient against the "MMSC Users" list is not performed.)*

Next, the MMSC checks all of the "MMSC Routing" definitions for a matching recipient address mask. Each route can have a list of recipient address masks associated with it, and if a match is found for a route, the message is routed via that route.

If no matching route is found, then the "Default Route" is used.

The "Default Route" can specify "Direct Delivery" utilizing the internal MMSC, "Convert MMS Message to Multimedia WAP Push", or it can specify any of the other defined "MMSC Routing".

For example, if you wanted to setup NowSMS so that it routed all outbound MMS messages to an operator MMSC over a GPRS modem connection, you would first add this routing to the "**MMSC Routing**" list, and the return to the "**Default Route**" prompt to select that routing as the default route for the system.

By contrast, if NowSMS was installed as an operator MMSC, you might want to define the "Default Route" to be a route that converts the MMS message to an SMS with a web link. Assuming that every time a new user sends a message via the MMSC, that user is automatically registered with the MMSC *(see the Technical Bulletin - Now MMSC Operator Considerations, beginning on page 363, for more information on this type of configuration)*, the MMSC would use direct delivery for users who were defined to the MMSC, and it would convert the MMS messages to an SMS with a web link for any recipients that were not yet registered with the MMSC.

MMSC Routings can use any of the following MMSC connectivity protocols:

MM7 – An XML/SOAP based format for MMS messages to be transmitted using HTTP POST. This standard is defined by the 3GPP. NowSMS also supports non-standard variations of MM7 that are used by the Ericsson, Materna AnnyWay and LogicaCMG MMSCs.

MM4 – An SMTP based format for MMS messages to be transmitted between MMSCs.

MM1 – A binary format for transmitting MMS messages using HTTP POST. This is the protocol that is used for phone to phone MMS, so if you are routing messages to an operator MMSC over a GPRS connection, this is the protocol that is used.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

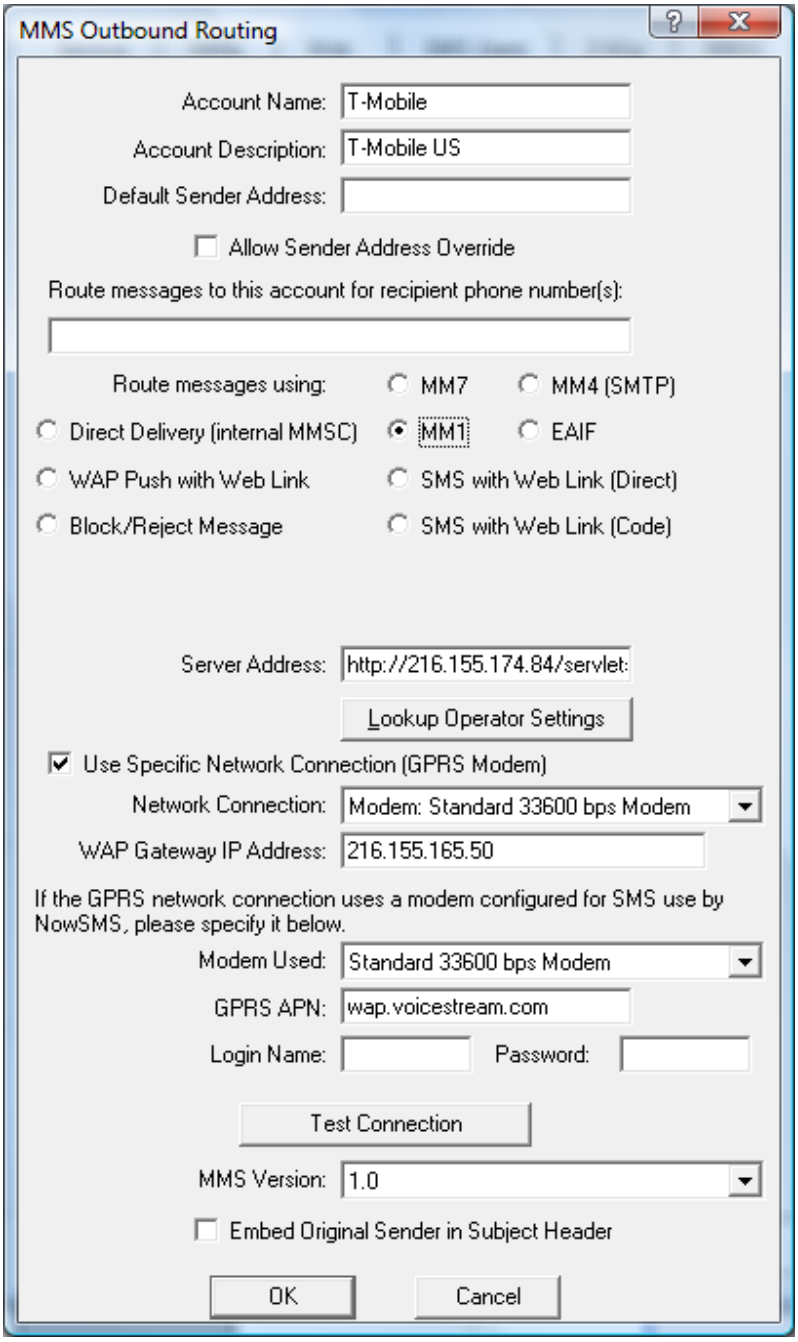
EAIF – This is a Nokia proprietary variation on the MM1 format that is used for sending messages to a Nokia MMSC.

Additionally, routes can be defined which convert MMS messages to multimedia WAP Push, or to SMS messages with a web link. Or a route can be defined for rejecting messages *(if the MMSC does not want to support delivery to external recipients)*.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending MMS Message – MM1 (GPRS Modem)

When adding an "MMSC Routing", the options on the dialog box will change, depending on the type of MMSC connection. Below is shown the configuration for an "MMSC Routing" using an MM1 connection using a GPRS modem:



"Account name" and **"Account Description"** are settings that identify the connection to the Now SMS/MMS Gateway only. These settings are not transmitted externally.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"Default Sender Address" specifies the default sender address to applied to any MMS messages that are transmitted over this connection. The "Default Sender Address" is used only if the message being transmitted does not include a sender address. This address can either be a standard e-mail address, or a telephone number. If a telephone number is specified, it must be specified in MMS messaging format (+phonenumber/TYPE=PLMN). If "Allow Sender Address Override" is checked, then submitted messages can include their own sender address. If this setting is not checked, the sender address for all messages transmitted via the MMSC connection will be changed to the "Default Sender Address".

When sending to an operator MMSC over GPRS, the operator MMSC will usually not allow a sender address other than the MSISDN of the sending phone. In these cases, it is usually best to leave the "Sender Address" field blank, and allow the MMSC to assign it automatically. To do this, leave the **"Default Sender Address"** field blank, and uncheck **"Allow Sender Address Override"**.

The **"Route messages to this account for recipient phone number(s)"** field is an address mask for defining which recipient phone numbers should be routed to this account. For example, "+44*" would route all messages for the UK country code (44) to this connection. Multiple address masks can be defined. When multiple address masks are defined, they should be separated by a comma only (,) and no white space characters.

If this field is left blank, no messages will be routed to this connection, except under special circumstances.

The **"Route Messages Via"** option defines the protocol that will be used for routing the MMS message. Messages can be routed to an external host using the MM7, MM4, MM1 or EAIF protocols. Other options for message routing include **"Direct Delivery"** via the gateway's internal MMSC, or an option for the MMS to be converted into a Multimedia WAP Push message to be received by the WAP client on the recipient phone instead of the MMS client.

For MM1 connections, the **"Server Address"** field should contain a URL for posting to the MMSC. (The "http://" portion of the address is optional.) For example, http://192.168.1.1:8080/mm1. Note that when you are using MM1 to connect with an operator MMSC over GRPS, this "Server address" is the "MMS Server URL" that would normally be configured on a mobile phone to connect with the operator MMSC.

If the MMS message content needs to be sent over a specific network connection, such as a GPRS modem, check the **"Use Specific Network Connection (GPRS Modem)"** checkbox.

In the **"Network Connection"**, select the name of the network connection that is to be used. NowSMS can use any of three different types of connections to make a GPRS connection. The different available connections are listed in the drop-down field associated with this configuration field. The different types of connections are prefixed with the text **"Modem:"**, **"Dial-up:"**, or **"Network:"**, and are described below:

a.) **"Modem:"** refers to a standard GPRS connection to be initiated over a GPRS modem. Select the modem that should be used for this connection. (Note that only modems that have a Windows modem driver defined for the modem can be used. If your modem does not have a modem driver supplied by the manufacturer, you can use one of the "Standard" or "Generic" modem drivers available when defining a modem in the Windows Control Panel.)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

b.) "**Dial-up:**" refers to a dial-up networking connection defined on the current PC. This setting can be used if you have advanced requirements and wish to create a custom dial-up networking profile.

c.) "**Network:**" refers to a particular network interface card installed in the PC. Some PC card GPRS modems, such as the Sierra Wireless Aircard 750 provide GPRS access via a network driver interface. To tell NowSMS to use that specific network driver for connecting to the MMSC, select the named "Network:" driver.

The "**WAP Gateway IP Address**" field should contain the IP address of the operator WAP gateway which will act as a proxy for connections to the MMSC. If the gateway uses WAP2/HTTP instead of WSP, prefix the address with "http://", using the format "http://ip.address:port".

For most GSM/GPRS modems, it is not possible for the Now SMS/MMS Gateway to connect to the modem to send/receive SMS messages as the same time as a GPRS connection is active (Network Connection type = "Modem:" or "Dial-up:"). For these situations, the "**Modem Used**" setting should specify the name of a modem that is used by the Dial-up networking profile. If the Now SMS/MMS Gateway is using the modem to send/receive SMS messages, it will automatically release the modem when it needs to initiate a GPRS connection, otherwise the GPRS connection will not be able to be properly established.

The "**GPRS APN**" field specifies the GPRS Access Point Name (APN) to be accessed for connecting to the MMSC. This setting is operator dependent, and it may be advisable to check the MMS configuration settings on a working mobile phone to determine the correct settings. Note that your mobile operator possibly has multiple GPRS APNs and multiple WAP gateways, and you need the settings that are appropriate for MMS, not for WAP browsing or general internet connectivity. Note that this setting is only available when using a Network Connection of type "Modem:". For other connection types, the GPRS APN must be configured for the connection using some configuration means external to NowSMS.

The "**Login Name**" and "**Password**" parameters specify a username and password to be used for connecting to the GPRS network.

The "**Test Connection**" dialog verifies that the Now SMS/MMS Gateway can make a TCP/IP connection to the specified "Server Address". Or, when an MM1 connection is used with a dial-up connection ("Use Specific Network Connection (GPRS Modem)"), the gateway will initiate a network connection to the specified profile, and that it can make a connection to the specified WAP gateway over the connection, and the "Server Address" is not tested at this time.

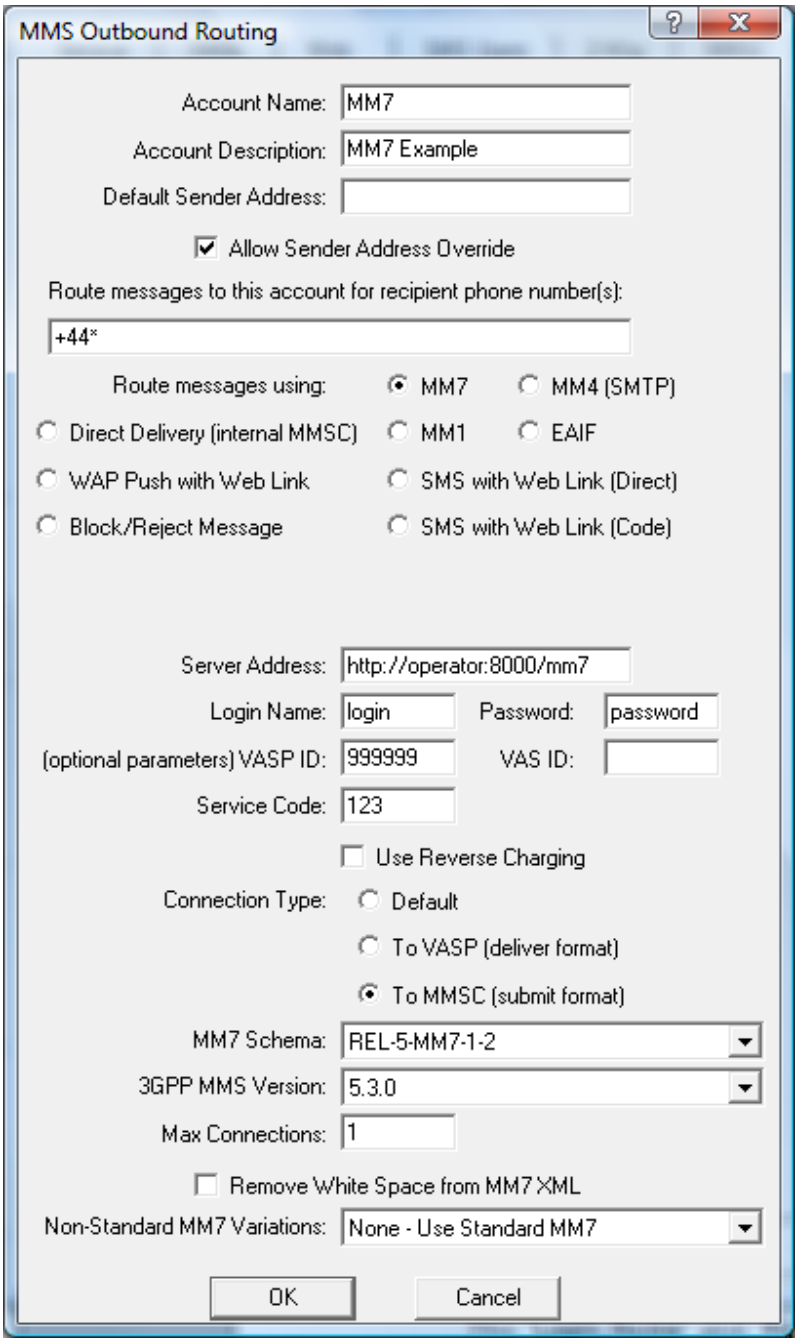
"**MMS Version**" specifies the version number of the OMA MMS Encapsulation Protocol to use when communicating with the operator MMSC. Note that support for read receipts requires version 1.2.

The "**Embed Original Sender in Subject Header**" is a special setting that can be used when NowSMS is operating as an E-Mail to MMS gateway with MMS messages being sent via a GPRS modem. When this option is checked, the original message sender is preserved in the subject header of the MMS message, so that if a reply is received back from the message recipient, NowSMS can route the reply to the original sender. For more information on this type of configuration, see **E-Mail to SMS/MMS Connectivity** on page 243.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending MMS Message – MM7

Below is shown the configuration for an "MMSC Routing" using an MM7 connection to an operator MMSC (or other MMSC provider):



"Account name" and **"Account Description"** are settings that identify the connection to the Now SMS/MMS Gateway only. These settings are not transmitted externally. **"Default Sender Address"** specifies the default sender address to applied to any MMS messages that are transmitted over this connection. The "Default Sender Address" is used

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

only if the message being transmitted does not include a sender address. This address can either be a standard e-mail address, or a telephone number. If a telephone number is specified, it must be specified in MMS messaging format (+phonenumber/TYPE=PLMN). If "Allow Sender Address Override" is checked, then submitted messages can include their own sender address. If this setting is not checked, the sender address for all messages transmitted via the MMSC connection will be changed to the "Default Sender Address".

When sending to an operator MMSC over GPRS, the operator MMSC will usually not allow a sender address other than the MSISDN of the sending phone. In these cases, it is usually best to leave the "Sender Address" field blank, and allow the MMSC to assign it automatically. To do this, leave the "**Default Sender Address**" field blank, and uncheck "**Allow Sender Address Override**".

The "**Route messages to this account for recipient phone number(s)**" field is an address mask for defining which recipient phone numbers should be routed to this account. For example, "+44*" would route all messages for the UK country code (44) to this connection. Multiple address masks can be defined. When multiple address masks are defined, they should be separated by a comma only (,) and no white space characters.

If this field is left blank, no messages will be routed to this connection, except under special circumstances.

For MM7 connections, the "**Server Address**" field should contain a URL for posting to the MMSC. (The "http://" portion of the address is optional.) For example, http://192.168.1.1:8080/mm7. The "**Login Name**" and "**Password**" specify an optional login name and password that will be used to login to the MMSC using HTTP Basic Authentication.

"**VASP ID**", "**VAS ID**" and "**Service Code**" values are optional parameters that may or may not be required by the MM7 provider. Your provider should indicator whether or not these parameters are required.

Check "**Use Reverse Charging**" if the recipient is to be charged for messages sent via this connection. (This is not supported by all MMSC providers.)

"**Connection Type**" should be "To MMSC (submit format)" if you are submitting messages to an operator or provider MMSC. "Connection Type" should be "To VASP (deliver format)" if you are configuring an MM7 connection that delivers message to an application.

"**MM7 Schema**" controls the MM7 XML Schema that is used when generating MM7 responses. Set this value only if the VASP requires a specific MM7 schema.

"**3GPP MMS Version**" controls the MMS Version that the MMSC uses when generating MM7 responses. Set this value only if the VASP requires a specific MMS version setting.

"**Max Connections**" specifies the maximum number of concurrent connections that NowSMS can make to this MMSC simultaneously in order to speed up message submission. (The default is 1.)

"**Remove White Space from MM7 XML**" - The MMSC normally generates MM7 XML in a user friendly format that includes line breaks. Some applications do not like any white space or line breaks within the MM7 XML, and this setting forces any white space to be removed from the XML.

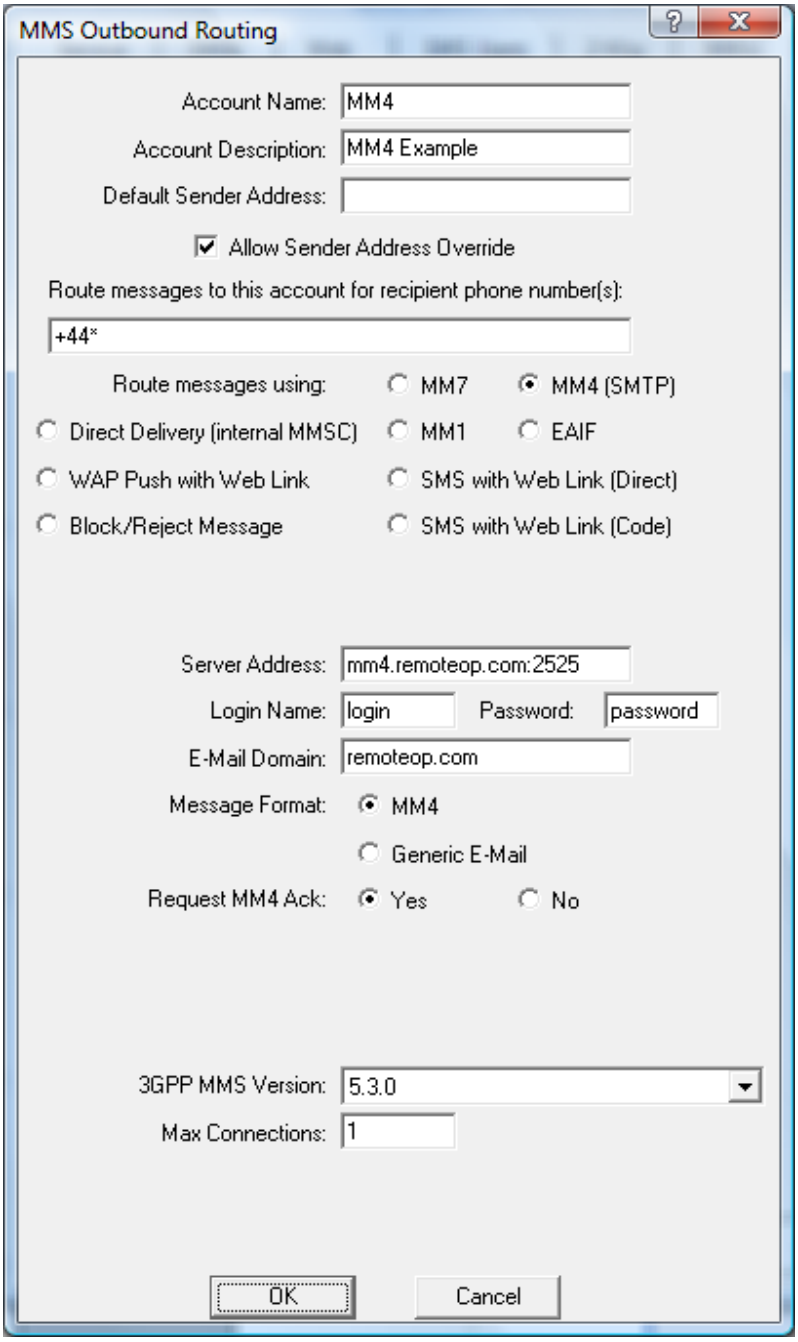
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"**Non-Standard MM7 Variations**" - This setting enables support for connecting to MMSCs that do not support the MM7 standard. NowSMS supports the non-standard MM7 variations deployed by Ericsson, Materna AnnyWay and LogicaCMG. Additionally, NowSMS supports an HTTP multipart file upload interface which can be useful for integrating with custom PHP scripts. This HTTP multipart file upload interface is described in more detail in **Receiving MMS Messages with a PHP Script: HTTP File Upload Post** on page 391.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending MMS Message – MM4

Below is shown the configuration for an "MMSC Routing" using an MM4 connection to another MMSC:



"Account name" and **"Account Description"** are settings that identify the connection to the Now SMS/MMS Gateway only. These settings are not transmitted externally. **"Default Sender Address"** specifies the default sender address to applied to any MMS messages that are transmitted over this connection. The "Default Sender Address" is used

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

only if the message being transmitted does not include a sender address. This address can either be a standard e-mail address, or a telephone number. If a telephone number is specified, it must be specified in MMS messaging format (+phonenumber/TYPE=PLMN). If "Allow Sender Address Override" is checked, then submitted messages can include their own sender address. If this setting is not checked, the sender address for all messages transmitted via the MMSC connection will be changed to the "Default Sender Address".

When sending to an operator MMSC over GPRS, the operator MMSC will usually not allow a sender address other than the MSISDN of the sending phone. In these cases, it is usually best to leave the "Sender Address" field blank, and allow the MMSC to assign it automatically. To do this, leave the "**Default Sender Address**" field blank, and uncheck "**Allow Sender Address Override**".

The "**Route messages to this account for recipient phone number(s)**" field is an address mask for defining which recipient phone numbers should be routed to this account. For example, "+44*" would route all messages for the UK country code (44) to this connection. Multiple address masks can be defined. When multiple address masks are defined, they should be separated by a comma only (,) and no white space characters.

If this field is left blank, no messages will be routed to this connection, except under special circumstances.

For MM4 connections, the "**Server Address**" field should contain an IP address or host name of the MMSC. This field can also include a port number in the format, hostname:port, where "25" is the default port if a port number is not explicitly specified. The "**E-Mail Domain**" field specifies an e-mail domain that should automatically be appended to phone numbers when routing via this MMSC connection.

"**Login Name**" and "**Password**" are optional parameters that should be used if the MMSC is to use "SMTP Authentication" when transferring messages using this MM4 connection.

"**Message Format**" should be set to "MM4" when connecting to another MMSC. It can be set to "Generic E-Mail" for situations where the connection is not using the true MM4 protocol *(this option omits MM4 specific headers which could cause confusion for connections that are not true MMS)*.

"**Request MM4 Ack**" specifies whether or not an acknowledgment of receipt should be requested from the receiving operator (this is not an acknowledgment of final message delivery, only an acknowledgment that the MMSC has received the message via MM4). Normally, this setting should be enabled, however it may be disabled if there are problems receiving acknowledgements from the other operator or service provider.

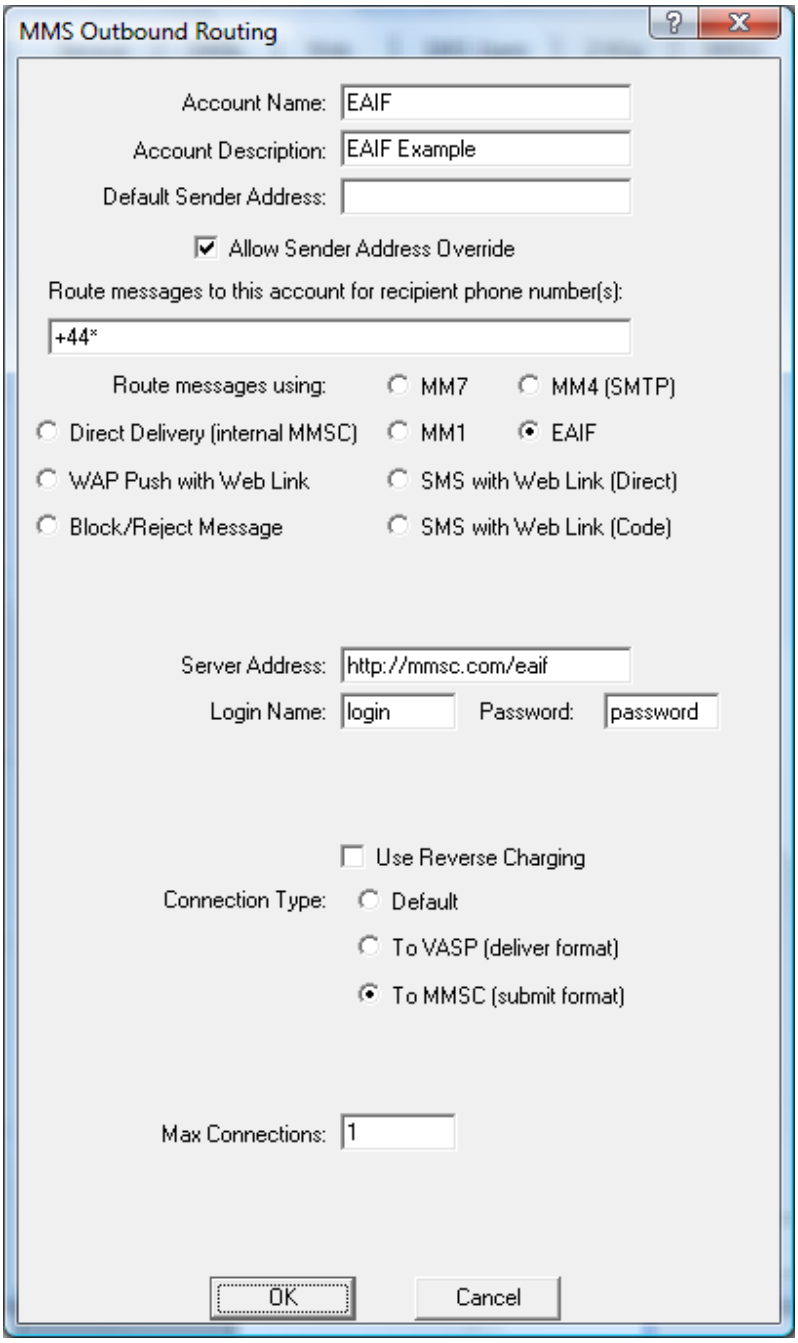
"**3GPP MMS Version**" controls the MMS Version that the MMSC uses when generating MM7 responses. Set this value only if the VASP requires a specific MMS version setting.

"**Max Connections**" specifies the maximum number of concurrent connections that NowSMS can make to this MMSC simultaneously in order to speed up message submission. (The default is 1.)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending MMS Message – EAIF

Below is shown the configuration for an "MMSC Routing" using an EAIF connection to another MMSC:



"Account name" and **"Account Description"** are settings that identify the connection to the Now SMS/MMS Gateway only. These settings are not transmitted externally.
"Default Sender Address" specifies the default sender address to applied to any MMS messages that are transmitted over this connection. The "Default Sender Address" is used

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

only if the message being transmitted does not include a sender address. This address can either be a standard e-mail address, or a telephone number. If a telephone number is specified, it must be specified in MMS messaging format (+phonenumber/TYPE=PLMN). If "Allow Sender Address Override" is checked, then submitted messages can include their own sender address. If this setting is not checked, the sender address for all messages transmitted via the MMSC connection will be changed to the "Default Sender Address".

When sending to an operator MMSC over GPRS, the operator MMSC will usually not allow a sender address other than the MSISDN of the sending phone. In these cases, it is usually best to leave the "Sender Address" field blank, and allow the MMSC to assign it automatically. To do this, leave the "**Default Sender Address**" field blank, and uncheck "**Allow Sender Address Override**".

The "**Route messages to this account for recipient phone number(s)**" field is an address mask for defining which recipient phone numbers should be routed to this account. For example, "+44*" would route all messages for the UK country code (44) to this connection. Multiple address masks can be defined. When multiple address masks are defined, they should be separated by a comma only (,) and no white space characters.

If this field is left blank, no messages will be routed to this connection, except under special circumstances.

For EAIF connections, the "**Server Address**" field should contain a URL for posting to the MMSC. (The "http://" portion of the address is optional.) For example, http://192.168.1.1:8080/eaif. The "**Login Name**" and "**Password**" specify an optional login name and password that will be used to login to the MMSC using HTTP Basic Authentication.

Check "**Use Reverse Charging**" if the recipient is to be charged for messages sent via this connection. (This is not supported by all MMSC providers.)

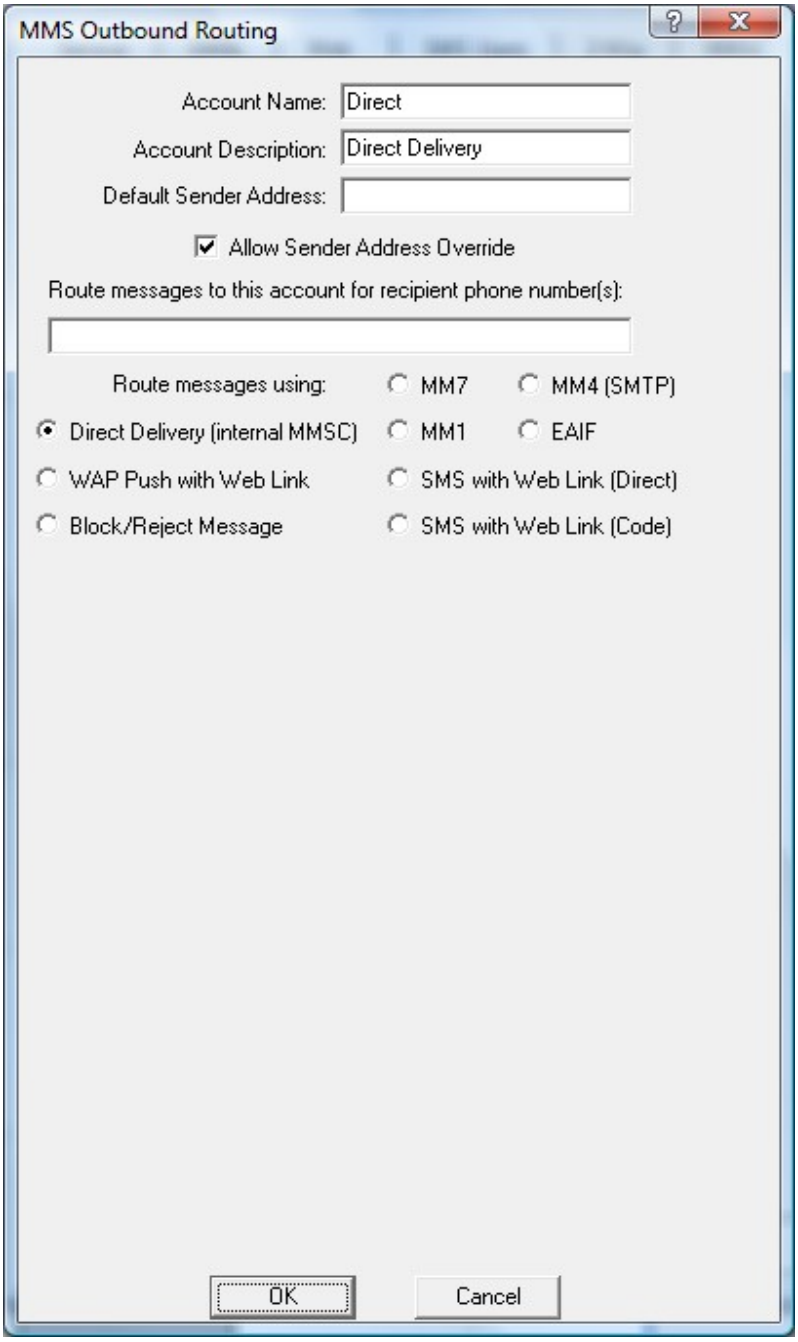
"**Connection Type**" should be "To MMSC (submit format)" if you are submitting messages to an operator or provider MMSC. "Connection Type" should be "To VASP (deliver format)" if you are configuring an EAIF connection that delivers message to an application.

"**Max Connections**" specifies the maximum number of concurrent connections that NowSMS can make to this MMSC simultaneously in order to speed up message submission. (The default is 1.)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending MMS Message – Direct Delivery

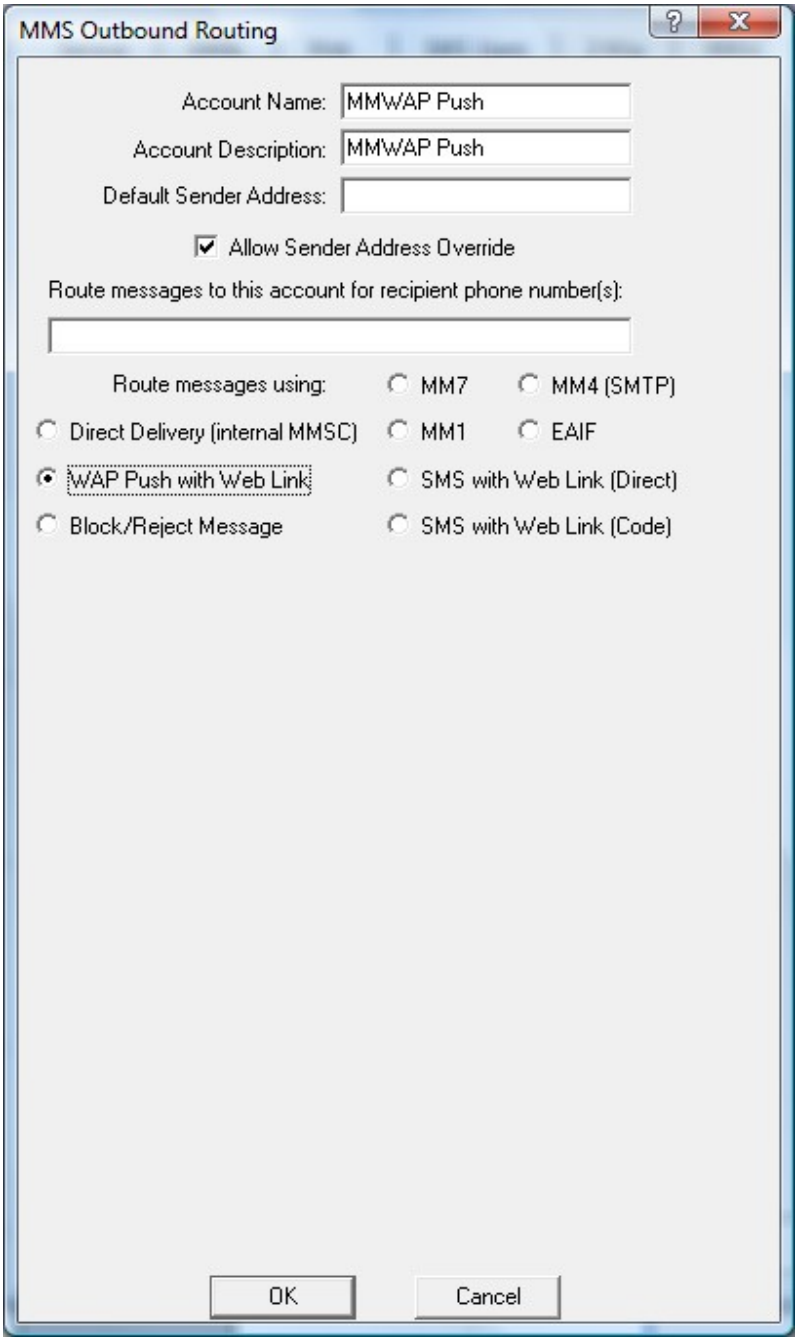
There are no parameters to define, other that "Account Name", "Account Description", and "Route Messages to this account for recipient phone number(s)", when defining a "Direct Delivery" route.



This routing option exists primarily to allow recipient address masks to be defined for recipient phone numbers for which the MMSC should perform MMS direct delivery.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending MMS Message – Convert to WAP Push with Web Link



When this route is applied, MMS messages will be converted into Multimedia Content Push messages. NowSMS automatically formats the MMS message content so that it is accessible via a dynamically generated URL on the MMSC, and sends a WAP Push message to the recipient. This WAP Push message includes a direct URL link to allow the content to be retrieved by the receiving device.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending MMS Message – Convert to SMS with Web Link

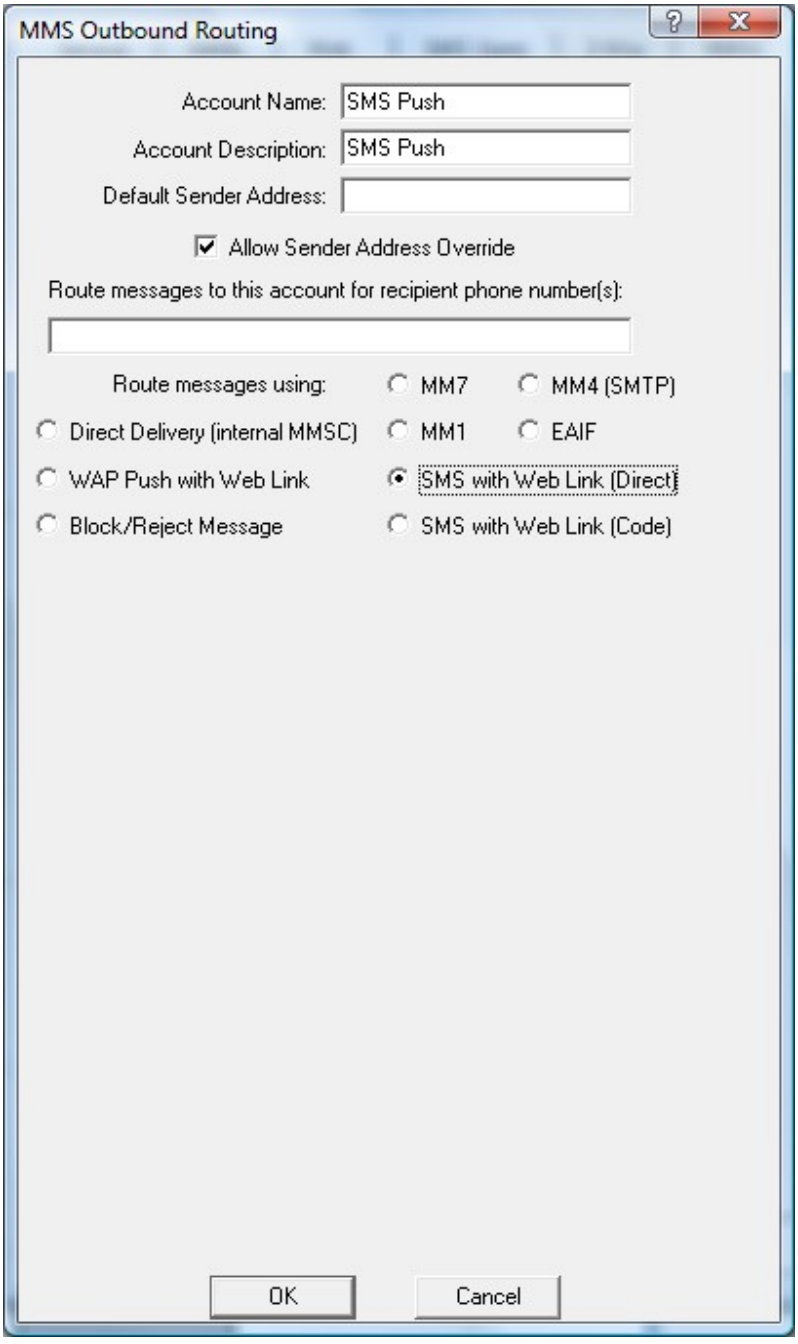
NowSMS supports two different "**SMS with Web Link**" options, which can lead to some confusion.

The ***"SMS with Web Link (Direct)" interface is designed to be used by content providers*** who wish to deliver multimedia content to mobile devices. The URL link in the SMS text message links directly to the message content, with the expectation that the recipient will retrieve content directly from the handset.

The ***"SMS with Web Link (Code)" interface is designed to be used by mobile operator MMSCs*** for delivering MMS message content to mobile devices that do not support MMS, or for supporting international delivery of MMS message content to destinations where there is no available MMS interconnection. The URL link in the SMS text message is a login page where the recipient must enter their phone number and a 4 digit code in order to access the message. The expectation is that the recipient is more likely to retrieve the content using a PC browser, although it is also possible to use a mobile phone based browser.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending MMS Message – Convert to SMS with Web Link (Direct)



When this route is applied, MMS messages will be converted into Multimedia Content Push messages. NowSMS automatically formats the MMS message content so that it is accessible via a dynamically generated URL on the MMSC, and sends an SMS text message to the recipient. This text message includes a direct URL link to allow the content to be retrieved by the receiving device using either a WAP/WML or HTML based browser.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending MMS Message – Convert to SMS with Web Link (Code)

The "SMS with Web Link (Code)" routing defines that any MMS messages should be converted to an SMS message that includes a link to a web page where the recipient can go to retrieve the content of the MMS message.

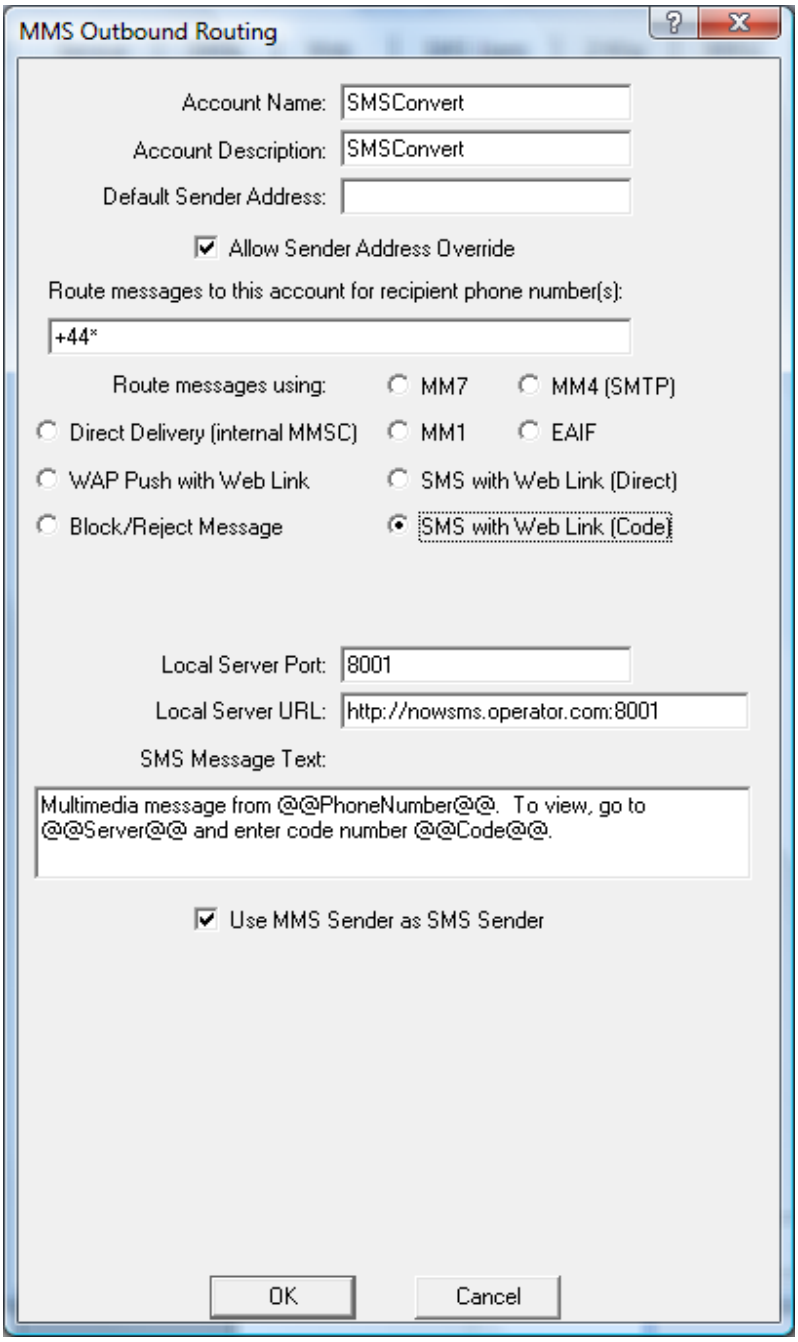
For example, when sending an MMS message to a recipient that does not have an MMS compatible phone, or to an international recipient to which there is no available MMS interconnection, the recipient can instead receive an SMS message similar to the following:

```
Multimedia message from +4477777777.  To view, go to
http://mms.domain.com:8080, and enter code number 1234.
```

The recipient can then navigate to the web link using a WAP browser on a mobile phone, or a web browser on a PC. They will be prompted for their phone number and the code number that was supplied in the SMS message. After entering that information, the message content will be displayed in the browser.

This route is frequently configured as a default route, which means that this route is used if a recipient was not in the "MMSC Users" list, and not covered by a recipient address mask in another "MMSC Routing" definition.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



The following configuration settings are required for this type of routing:

"**Local Server Port**" specifies an available port number on the PC running NowSMS which will be used to accept connections from recipients who are attempting to connect in to retrieve MMS messages over the web interface. This port number must be different from other port numbers configured for use by NowSMS. This port must be unique, because the only functionality provided through this interface is MMS message retrieval over the web interface. Other NowSMS ports will likely have restricted access via a firewall, however this port needs to be open to the outside world to allow these types of messages to be retrieved.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"Local Server URL" specifies the externally accessible URL that recipients will access to connect to the "Local Server Port" on the MMSC. This setting will default to the "Local Host Name or IP Address" configured for the MMSC, and the "Local Server Port". However, if you are remapping the address and port via a firewall, you should specify the external host name (and port if required) in this field. Keep in mind that some users may be restricted from retrieving content from non-standard web server ports.

"SMS Message Text" is a template for the SMS text message that will be sent out to recipients. This text should consist only of characters that are part of the default GSM character set. And it should include the following replaceable text parameters:

@@PhoneNumber@@ will be replaced with the phone number of the message sender.

@@Server@@ will be replaced with the value configured for "Local Server URL".

@@Code@@ will be replaced with a code number that the recipient must enter in order to retrieve the MMS message content.

The default text for this message is:

```
"Multimedia message from @@PhoneNumber@@.  To view, go to @@Server@@
and enter code number @@Code@@."
```

"Use MMS Sender as SMS Sender" - This option specifies that the SMS message that is sent out should use the sender address from the original MMS sender, if the original MMS sender was a phone number.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending MMS Message – Block/Reject Message

There are no parameters to define, other that "Account Name", "Account Description", and "Route Messages to this account for recipient phone number(s)", when defining a "Block/Reject Message" route.

When this route is applied, the MMSC will block/reject MMS messages to any recipient phone numbers that match the recipient address mask defined in the "Route via" parameter.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Connecting to an Operator MMSC – Receiving MMS Messages

This section describes configuration information specific to receiving MMS messages from an operator MMSC using MM7, MM4 or EAIF. Receiving MMS messages over an MM1 connection using a GPRS modem is described in the section titled **Connecting to an Operator MMSC – Using a GPRS Modem** *(page 136)*.

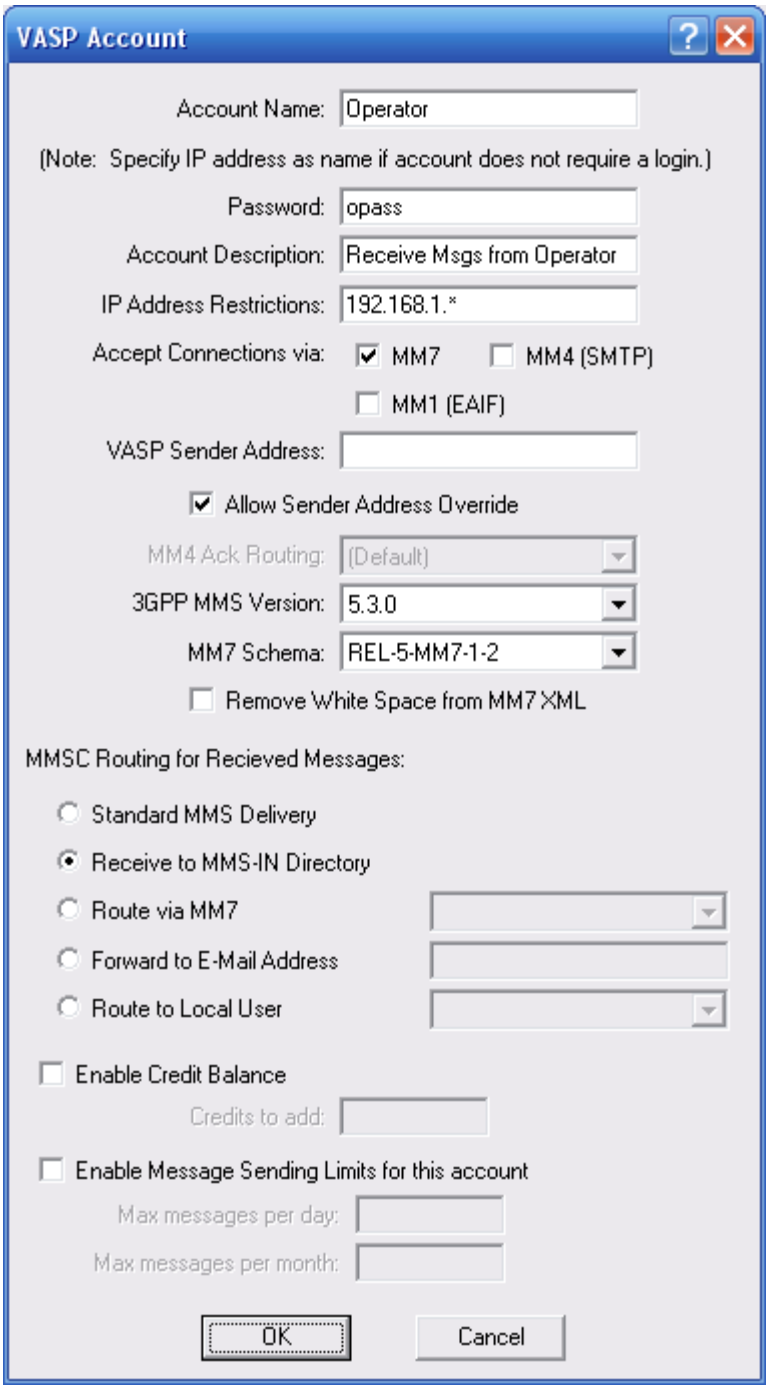
In order to receive messages from an operator MMSC using MM7, MM4 or EAIF, the mobile operator must configure their MMSC to initiate outbound connections to your MMSC for message delivery.

The process of configuring the Now SMS/MMS Gateway to send messages to an operator MMSC is separate from the process of configuring the gateway to receive messages from an operator MMSC.

While an operator MMSC is technically not a VASP (Value Added Service Provider), in order to receive MMS messages from an operator MMSC, you must configure a VASP account for the mobile operator on the **"MMSC VASP"** configuration dialog. Please refer to the section titled **Configuring MMS VASP Accounts** *(page 132)* for more information on these configuration settings.

When an operator MMSC connects to your system to deliver received MMS messages, the process is the same as a VASP connecting to your system to submit messages for external delivery. The primary difference is that the **"MMSC Routing for Received Messages"** setting is different for a VASP compared to an operator MMSC connection.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



When defining a VASP account that is used to receive messages from a mobile operator, it is important that the "**MMSC Routing for Received Messages**" be set to **any value other than** "**Standard MMS Delivery**". This is because the "Standard MMS Delivery" option would specify that the messages should be routed for external delivery. However, when receiving messages from an operator MMSC, you would want to perform receive processing against the messages.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Submitting MMS Messages to NowSMS

Once NowSMS has been successfully configured to send MMS messages via the web interface, it is possible to use various APIs to sending MMS messages programmatically.

The NowSMS APIs for sending MMS messages are based upon either the HTTP or SMTP protocols. However, before we go into details about these protocols, there may be an easy way to interface with NowSMS without learning the details of these protocols.

If you work with PHP, there is an example PHP script for sending MMS messages that is described on page 166 under the header **Send MMS Message with PHP**.

If you work with Java, there is a Java class for sending MMS messages that is described on page 172 under the header **Send MMS Message with Java**.

There is also a command-line interface for Windows environments which allows you to send MMS messages by spawning a command line script from your application. This script is described on page 176 under the header **Send MMS Message from Command Line**.

Applications can also submit MMS messages directly to the Now SMS/MMS Gateway via any of the following protocols, which are described below:

1.) **Now SMS/MMS Proprietary URL Submission** via HTTP GET or POST. *(See page 178)*
2.) **MM7** – The MMS standard for applications to submit MMS messages to an MMSC. MM7 is an XML/SOAP based API where the MMS message is formatted in a MIME encoded XML document and posted to the server using an HTTP POST. *(See page 180)*
3.) **MM4** – The MMS standard for connectivity between multiple MMSCs. MM4 is an SMTP based interface where the MMS message is posted to the server as a standard MIME encoded e-mail message. While the interface exists primarily for connectivity between multiple MMSCs, the Now SMS/MMS Gateway also makes this interface available to application developers that are more comfortable with the SMTP protocol. *(See page 183)*
4.) **MM1** – The MMS standard for phones to send and receive MMS messages from an MMSC. This is an HTTP based protocol where applications can submit binary encoded MMS messages encoding according to the MMS Encapsulation Specification (application/vnd.wap.mms-message MIME type) to the gateway using HTTP POST. *(See page 185)*
5.) **EAIF** – This is a Nokia proprietary variation on the MM1 protocol which was defined as an interface for submitting messages to a Nokia MMSC. The interface is functionally similar to MM1, with additional HTTP headers defined. *(See page 186)*

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send MMS Message with PHP

The following PHP script (sendmms.php) uses the Now SMS/MMS Proprietary URL Submission interface to send MMS messages.

<?php

```
/* Function to perform HTTP POST of multi-part data */

function MmsSend ($host, $port, $username, $password, $data_to_send)
{
    $dc = 0;
    $bo ="----------------------------mime-boundary-marker";

    $fp = fsockopen($host, $port, $errno, $errstr);
    if (!$fp) {
        echo "errno: $errno \n";
        echo "errstr: $errstr\n";
        return $result;
    }

    fputs($fp, "POST / HTTP/1.1\r\n");
    if ($username != "") {
       $auth = $username . ":" . $password;
       $auth = base64_encode($auth);
       fwrite($fp, "Authorization: Basic " . $auth . "\r\n");
    }
    fputs($fp, "User-Agent: NowSMS PHP Script\r\n");
    fputs($fp, "Accept: */*\r\n");
    fputs($fp, "Content-type: multipart/form-data; boundary=$bo\r\n");

    foreach($data_to_send as $key=>$val) {
        $ds =sprintf("%s\r\nContent-Disposition: form-data; name=\"%s\"\r\n%s\r\n",$bo,$key,$val);
        $dc += strlen($ds);

    }
    $dc += strlen($bo)+3;
    fputs($fp, "Content-length: $dc\r\n");
    fputs($fp, "\r\n");
    fputs($fp, "This is a MIME message\r\n\r\n");

    foreach($data_to_send as $key=>$val) {
        $ds =sprintf("%s\r\nContent-Disposition: form-data; name=\"%s\"\r\n%s\r\n",$bo,$key,$val);
        fputs($fp, $ds );
    }
    $ds = $bo."--\r\n" ;
    fputs($fp, $ds);

    $res = "";

    while(!feof($fp)) {
        $res .= fread($fp,1);
    }
    fclose($fp);


    return $res;
}

/* Add a file part to the multipart data */
function MmsAddFile ($data, $file, $contenttype)
{

$fa = @file($file);
$xf ="Content-Type: ".$contenttype."\r\n\r\n".implode("",$fa);
$data["MMSFile\"; filename=\"$file"]   = $xf;

return $data;
}

/* Add a field to the multipart data */
function MmsAddField ($data, $fieldname, $fieldvalue) {

$data[$fieldname] = "\r\n" . $fieldvalue;
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
return $data;

}

/* Initialise the MMS message */
function MmsInit () {

   $data = "";
   return $data;
}


/* Set parameters for connecting to the NowSMS server */
$nowsmsHostName = "127.0.0.1";    /* IP Address or host name of NowSMS Server */
$nowsmsHostPort = "8800";         /* NowSMS Port number for the web interface */
$nowsmsUsername = "test";         /* "SMS Users" account name */
$nowsmsPassword = "test";         /* "SMS Users" account password

/* Initialise the MMS Message structure */
$mmsMessage = MmsInit();

/* Set MMS message fields */
/* "PhoneNumber" is the recipient, and can be a comma deilmited list of recipients or the name of a
NowSMS distribution list */
/* "MMSFrom" is the sender */
/* "MMSSubject" is the subject */
/* "MMSText" is an optional text part of the message.  Text parts can also be added as file references
*/
/* For additional parameters, please see http://blog.nowsms.com/search/label/sendmms.php */
$mmsMessage = MmsAddField ($mmsMessage, "PhoneNumber", "+4477777777777");
$mmsMessage = MmsAddField ($mmsMessage, "MMSFrom", "sender@domain.com");
$mmsMessage = MmsAddField ($mmsMessage, "MMSSubject", "Subject of Message");
/* The MMSText field is optional */
$mmsMessage = MmsAddField ($mmsMessage, "MMSText", "Hello!");

/* Add the file parts here, referencing local files.
   Specify a path to the file, remembering to escape backslashes in the path (c:\temp\file becomes
c:\\temp\\file).
   The last parameter is the MIME content type, e.g., "image/gif", "image/jpeg", "image/png",
"text/plain" or "application/smil" ...
   however, note that current versions of NowSMS ignore the MIME content type when messages are
submitted via the interface
   used by this PHP script. Instead, NowSMS uses the file extension to determine the content type
(e.g., ".gif", ".jpg", ".png", ".txt", ".smil"
*/

$mmsMessage = MmsAddFile ($mmsMessage, "f:\\temp\\file.gif", "image/gif");

/* Now send the message.
   The HTTP response from the server is returned by this function, and this example echoes it to the
console. */

$x   = MmsSend ($nowsmsHostName, $nowsmsHostPort, $nowsmsUsername, $nowsmsPassword, $mmsMessage);
echo $x;


?>
```

A copy of the script can be downloaded at http://www.nowsms.com/download/sendmms-php.txt.

The first part of sendmms.php consists of PHP functions that you will call in your PHP script ... namely **MmsInit**, **MmsAddField**, **MmsAddFile** and **MmsSend**. Include these functions in your script without editing them.

After these functions are defined, sendmms.php contains a simple example showing how to use these functions to send an MMS message through a NowSMS server.

1.) First, you need to initialise the parameters to point to your NowSMS server:

```
/* Set parameters for connecting to the NowSMS server */
$nowsmsHostName = "127.0.0.1"; /* IP Address or host name of NowSMS Server */
$nowsmsHostPort = "8800"; /* NowSMS Port number for the web interface */
$nowsmsUsername = "test"; /* "SMS Users" account name */
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
$nowsmsPassword = "test"; /* "SMS Users" account password */
```

2.) Second, you need to call MmsInit to initialise the MMS message structure.

```
$mmsMessage = MmsInit();
```

3.) Third, you need to add the necessary MMS header fields and attributes desired for your MMS message, calling the MmsAddField fuction.

```
$mmsMessage = MmsAddField ($mmsMessage, "PhoneNumber", "+447777777777");
$mmsMessage = MmsAddField ($mmsMessage, "MMSFrom", "sender@domain.com");
$mmsMessage = MmsAddField ($mmsMessage, "MMSSubject", "Subject of Message");
```

The "**PhoneNumber**" field specifies the recipient(s) for the MMS message. This can be a comma delimited list of phone numbers, or it can be the name of a NowSMS distribution list.

The "**MMSFrom**" field specifies the sender of the MMS message. Normally, this would be a phone number, short code or e-mail address. (And your MMS service provider may either require a specific value here, or they may overwrite the value you supply with the address associated with your service.)

The "**MMSSubject**" field specifies the subject of the MMS message.

Those are the most common MMS header fields. Optionally, you might also want to include an "**MMSText**" field to specify some text to be included in the MMS message. Text can also be included in an MMS message as a text file reference.

4.) Fourth, you specify the files to include in the MMS message using the MmsAddFile function. These files might be images, video, text, or other file types supported by the receiving device.

```
$mmsMessage = MmsAddFile ($mmsMessage, "f:\\temp\\filename.gif", "image/gif");
```

An MMS message can contain one or more of these included files. If you do not include a SMIL file component, NowSMS will build one automatically, so for full control of the MMS message presentation, you will want to include your own SMIL file, where you reference your file components by their short filename (without the full path, e.g., filename.gif ... NOT f:\temp\filename.gif).

The files referenced in the PHP script must be local files, residing on the same server as the PHP script. Remember to escape backslashes in the path so as not to confuse the PHP interpreter(c:\temp\file becomes c:\\temp\\file).

The last parameter of MmsAddFile is the MIME content type, e.g., "image/gif", "image/jpeg", "image/png", "text/plain" or "application/smil" ... however, note that current versions of NowSMS ignore the MIME content type when messages are submitted via the interface used by this PHP script. Instead, NowSMS uses the file extension to determine the content type (e.g., ".gif", ".jpg", ".png", ".txt", ".smil").

5.) Fifth and finally, you call MmsSend to submit the MMS message.

```
MmsSend ($nowsmsHostName, $nowsmsHostPort, $nowsmsUsername, $nowsmsPassword, $mmsMessage);
```

That is the basic information required for using the sendmms.php script to send MMS messages.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The **MmsAddField** function can be used to specify any NowSMS URL parameter that is valid for sending an MMS message.

For example ... here is an incomplete list of additional parameter fields that can be specfied using the MmsAddField function.

"MMSFile" - As noted earlier, this script attaches local files to the MMS message. However, what if you want to include files that reside on a separate web server instead? In that case, do not use the MmsAddFile function. Instead, use $mmsMessage = MmsAddField ($mmsMessage, "MMSFile", "http://www.nowsms.com/media/logo.png"); and specify the file components as URL references via the "MMSFile" parameter field.

"MMSDeliveryReport" - "Delivery Report" specifies whether or not a delivery report is requested for the message. Set to "Yes" to request a delivery report. Note that any delivery report would be directed back to the phone number or e-mail address specified in the "MMSFrom" address.

"MMSReadReport" - "Read Report" specifies whether or not a read receipt is requested for the message. Note that the receiving client may choose not to send a read receipt. Any read receipt report would be directed back to the phone number or e-mail address specified in the "MMSFrom" address.

"MMSPriority" - "Priority" is a user defined priority to be associated with the message. Generally, any priority definition associated with the message is ignored by the underlying transport, but the receiving client may decide to display messages differently based upon this priority setting.

"MMSMessageClass" - "Message Class" is an attribute defined in the MMS specifications. "Personal" is the message type that is used for standard user-to-user communications. Other defined message classes that are supported by this parameter include: "Informational" and "Advertisement".

"MMSWAPPush" - Set to "Yes" to indicate that the message being sent should be sent as an "Multimedia Content Push" message via WAP Push instead of as an MMS message.

"MMSSMSPush" - Set to "Yes" to indicate that the message being sent should be sent as an "Multimedia Content Push" message via a URL link in a text SMS message instead of as an MMS message.

It is also possible to specify forward locking and DRM constraints to be applied against the content of the MMS message. Forward locking and DRM constraints apply to non-text parts of the MMS message (i.e., in a forward locked message, text could still be forwarded, but images or video could not). Please note that not all devices support forward locking and DRM constraints, therefore use these parameter settings only after testing thoroughly with mobile phones used by your message recipients.

"MMSForwardLock" - Forward locking is the most basic level of DRM (Digital Rights Management). When "Forward Lock" is set to "Yes", this indicates that the receiving device should not allow any non-text objects in the message to be forwarded off of the device. The device may allow the user to extract pictures, videos or sounds from the message and save them on the phone. However, any such objects remain forward locked, such that they cannot be forwarded to another user or transferred to another device. (IMPORTANT NOTE:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

NOT ALL DEVICES SUPPORT FORWARD LOCK, WHEN NOT SUPPORTED THE CONTENT WILL APPEAR AS GARBAGE OR MAY BE REJECTED BY THE OPERATOR MMSC.)

"DRMRestrict" - Beyond forward locking, More advanced DRM (Digital Rights Management) restrictions can be applied to limit the number of times that the user can access an object, or start and end dates can be specified to limit how long the user can access an object.

These advanced DRM restrictions can be applied by setting "DRMRestrict" to "Yes". When this setting is enabled, forward lock is also implied, and the value of the "MMSForwardLock" setting is ignored. (IMPORTANT NOTE: NOT ALL DEVICES SUPPORT DRM RESTRICTIONS, WHEN NOT SUPPORTED THE CONTENT WILL APPEAR AS GARBAGE OR MAY BE REJECTED BY THE OPERATOR MMSC.)

"DRMRestrictTextXML" - "Yes" specifies that the rights object should be encoded in text XML format. "No" specifies that the rights object should be encoded in binary XML format. The default is "No".

When DRM Restrictions are specified, it is generally necessary to specify one or more DRM Permissions and one or more DRM Constraints regarding the MMS message content.

DRM Permissions specify what types of access are allowed against the objects in a message that is protected with DRM.

For example, an audio or video object requires "Play" permission before the user can access it. An image requires "Display" permission before the user can access it, and it requires "Print" permission if the user is to be allowed to print the image to a printer , perhaps over Bluetooth. An application requires "Execute" permission before the user can make use of the application. In all cases, the forward locking is assumed, so that the user is not allowed to forward or transfer the object from the device.

If you are sending multiple types of objects in the MMS message, specify all permissions that are required for the different types of objects that are being sent.

"DRMPermissionPlay" - Set to "Yes" to enable DRM "Play" Permission.

"DRMPermissionDisplay" - Set to "Yes" to enable DRM "Display" Permission.

"DRMPermissionExecute" - Set to "Yes" to enable DRM "Execute" Permission.

"DRMPermissionPrint" - Set to "Yes" to enable DRM "Print" Permission.

DRM Constraints specify constraints with regard to how long a DRM protected object object should remain accessible to the user.

"DRMConstraintCount" - "# of Accesses (count)" specifies the the user can only access the DRM protected object this number of times before access is no longer allowed.

"DRMConstraintStart" - "Start Date (yyyy-mm-dd)" specifies that the user will not be allowed to access the DRM protected object until on or after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2008-12-24.)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"DRMConstraintEnd" - "End Date (yyyy-mm-dd)" specifies that the user will not be allowed to access the DRM protected object after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2008-02-24.)

"DRMConstraintInterval" - "# of Days (interval)" specifies that the user will be allowed to access the DRM protected object for this number of days after initial receipt of the object. The user can either enter a number of days here, or they can enter any valid value defined for the "" element in the OMA DRM Rights Expression Language specification. For example, P2Y10M15DT10H30M20S represents a duration of 2 years, 10 months, 15 days, 10 hours, 30 minutes and 20 seconds.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send MMS Message with Java

A Java example for sending MMS messages via NowSMS can be downloaded from the following link:

http://www.nowsms.com/download/sendmms.java.txt

This class supports all of the MMS related parameters available in NowSMS, most of which are described below.

To use this Java class, begin by creating a new sendmms object, specifying the address of the NowSMS server, and a valid username and password for a user account ("SMS Users") on the NowSMS Server.

```
sendmms mms = new sendmms ("http://127.0.0.1:8800/", "test", "test");
```

The addparameter method is used to build the MMS message object.

Start by specifying a recipient using the "**PhoneNumber**" parameter (this can be a comma delimited list of recipients):

```
mms.addparameter ("PhoneNumber", "+9999999999");
```

Next, add any desired MMS header parameters, such as the message subject using the "**MMSSubject**" parameter, or an optional text part of the message using the "**MMSText**" parameter:

```
mms.addparameter ("MMSSubject", "This a a test message");
mms.addparameter ("MMSText", "test message"); // Optional
```

Next, add the file objects for the MMS content using the "**MMSFile**" parameter.

```
mms.addparameter ("MMSFile", new File("f:\\temp\\test.jpg"));
mms.addparameter ("MMSFile", new File("f:\\temp\\test2.jpg"));
```

(Note: Remember to escape the "\" character as "\\" in your Java code.)

The send method submits the MMS message to NowSMS.

```
mms.send ();
```

The send method returns a string containing the MMS Message ID assigned for the submitted messages, in the following format:

```
MMSMessageID=xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
```

Note that a try/catch exception handler must be added around the object. Here's a complete example:

```
try {
sendmms mms = new sendmms ("http://127.0.0.1:8800/", "test", "test");
mms.addparameter ("PhoneNumber", "+9999999999");
mms.addparameter ("MMSSubject", "This a a test message");
mms.addparameter ("MMSText", "test message"); // Optional
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
mms.addparameter ("MMSFile", new File("f:\\temp\\test.jpg"));
mms.addparameter ("MMSFile", new File("f:\\temp\\test2.jpg"));
mms.send ();
}
catch(IOException e) {
System.out.println("unable to create new url: "+e.getMessage());
}
```

The mms.addparameter method can be used to specify any NowSMS URL parameter that is valid for sending an MMS message. For example ... here is an incomplete list of additional parameter fields that can be specfied using the mms.addparameter.

"MMSDeliveryReport" - "Delivery Report" specifies whether or not a delivery report is requested for the message. Set to "Yes" to request a delivery report. Note that any delivery report would be directed back to the phone number or e-mail address specified in the "MMSFrom" address.

"MMSReadReport" - "Read Report" specifies whether or not a read receipt is requested for the message. Note that the receiving client may choose not to send a read receipt. Any read receipt report would be directed back to the phone number or e-mail address specified in the "MMSFrom" address.

"MMSPriority" - "Priority" is a user defined priority to be associated with the message. Generally, any priority definition associated with the message is ignored by the underlying transport, but the receiving client may decide to display messages differently based upon this priority setting.

"MMSMessageClass" - "Message Class" is an attribute defined in the MMS specifications. "Personal" is the message type that is used for standard user-to-user communications. Other defined message classes that are supported by this parameter include: "Informational" and "Advertisement".

"MMSWAPPush" - Set to "Yes" to indicate that the message being sent should be sent as an "Multimedia Content Push" message via WAP Push instead of as an MMS message.

"MMSSMSPush" - Set to "Yes" to indicate that the message being sent should be sent as an "Multimedia Content Push" message via a URL link in a text SMS message instead of as an MMS message.

It is also possible to specify forward locking and DRM constraints to be applied against the content of the MMS message. Forward locking and DRM constraints apply to non-text parts of the MMS message (i.e., in a forward locked message, text could still be forwarded, but images or video could not). Please note that not all devices support forward locking and DRM constraints, therefore use these parameter settings only after testing thoroughly with mobile phones used by your message recipients.

"MMSForwardLock" - Forward locking is the most basic level of DRM (Digital Rights Management). When "Forward Lock" is set to "Yes", this indicates that the receiving device should not allow any non-text objects in the message to be forwarded off of the device. The device may allow the user to extract pictures, videos or sounds from the message and save them on the phone. However, any such objects remain forward locked, such that they cannot be forwarded to another user or transferred to another device. (IMPORTANT NOTE: NOT ALL DEVICES SUPPORT FORWARD LOCK, WHEN NOT SUPPORTED THE CONTENT WILL APPEAR AS GARBAGE OR MAY BE REJECTED BY THE OPERATOR MMSC.)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"DRMRestrict" - Beyond forward locking, More advanced DRM (Digital Rights Management) restrictions can be applied to limit the number of times that the user can access an object, or start and end dates can be specified to limit how long the user can access an object.These advanced DRM restrictions can be applied by setting "DRMRestrict" to "Yes". When this setting is enabled, forward lock is also implied, and the value of the "MMSForwardLock" setting is ignored. (IMPORTANT NOTE: NOT ALL DEVICES SUPPORT DRM RESTRICTIONS, WHEN NOT SUPPORTED THE CONTENT WILL APPEAR AS GARBAGE OR MAY BE REJECTED BY THE OPERATOR MMSC.)

"DRMRestrictTextXML" - "Yes" specifies that the rights object should be encoded in text XML format. "No" specfies that the rights object should be encoded in binary XML format. The default is "No".

When DRM Restrictions are specfied, it is generally necessary to specify one or more DRM Permissions and one or more DRM Constraints regarding the MMS message content.

DRM Permissions specify what types of access are allowed against the objects in a message that is protected with DRM.

For example, an audio or video object requires "Play" permission before the user can access it. An image requires "Display" permission before the user can access it, and it requires "Print" permission if the user is to be allowed to print the image to a printer , perhaps over Bluetooth. An application requires "Execute" permission before the user can make use of the application. In all cases, the forward locking is assumed, so that the user is not allowed to forward or transfer the object from the device.

If you are sending multiple types of objects in the MMS message, specify all permissions that are required for the different types of objects that are being sent.

"DRMPermissionPlay" - Set to "Yes" to enable DRM "Play" Permission.

"DRMPermissionDisplay" - Set to "Yes" to enable DRM "Display" Permission.

"DRMPermissionExecute" - Set to "Yes" to enable DRM "Execute" Permission.

"DRMPermissionPrint" - Set to "Yes" to enable DRM "Print" Permission.

DRM Constraints specify constraints with regard to how long a DRM protected object object should remain accessible to the user."DRMConstraintCount" - "# of Accesses (count)" specifies the the user can only access the DRM protected object this number of times before access is no longer allowed.

"DRMConstraintStart" - "Start Date (yyyy-mm-dd)" specifies that the user will not be allowed to access the DRM protected object until on or after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2008-12-24.)

"DRMConstraintEnd" - "End Date (yyyy-mm-dd)" specifies that the user will not be allowed to access the DRM protected object after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2008-02-24.)

"DRMConstraintInterval" - "# of Days (interval)" specifies that the user will be allowed to access the DRM protected object for this number of days after initial receipt of the object.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The user can either enter a number of days here, or they can enter any valid value defined for the "" element in the OMA DRM Rights Expression Language specification. For example, P2Y10M15DT10H30M20S represents a duration of 2 years, 10 months, 15 days, 10 hours, 30 minutes and 20 seconds.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send MMS Message from Command Line

A Windows command line script that enables sending MMS messages from the command line can be downloaded form http://www.nowsms.com/download/sendmms.vbs.txt.

Assuming that the script file is saved as a file named smsmms.vbs, you would issue the following command to send an MMS message:

```
cscript sendmms.vbs recipient[,recipient2,recipient3...] filename [filename2]
[filename3...] [variable=setting]
```

recipient can contain either one recipient phone number, or a comma delimited list of recipients, e.g., +44777777777,+44777777778 (important - do not include spaces)

filename is the name of a local file to be included in the MMS message. Similar to when submitting an MMS message via the "Send MMS Message" web form, you may specify multiple files that NowSMS will package up as an MMS message. (Note: NowSMS identifies content types based on the file extension of the submitted file, see the MMSCTYPE.INI file for the default list of file extension to MIME type mappings.)

variable=setting can specify additional URL parameters supported by NowSMS, such as:

MMSSubject=SubjectLine

MMSFrom=SenderAddress

MMSDeliveryReport=Yes

MMSReadReport=Yes

MMSPriority=High Normal Low

MMSMessageClass=Personal Informational Advertisement

MMSForwardLock=Yes

DRMRestrict=Yes

DRMRestrictTextXML=Yes

DRMPermissionPlay=Yes

DRMPermissionDisplay=Yes

DRMPermissionExecute=Yes

DRMPermissionPrint=Yes

DRMConstraintCount=##

DRMConstraintStart=yyyy-mm-dd

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

DRMConstraintEnd=yyyy-mm-dd

DRMConstraintInterval=##

Note: If the Subject line contains a space character, put quotes around the setting, e.g., "MMSSubject=This is a test message"

Examples:

```
cscript sendmms.vbs +4477777777 image.gif filename.txt "MMSSubject=This is a
test message"

cscript sendmms.vbs +4477777777,+4477777778 image.gif "MMSText=This is some
message text" MMSFrom=+44777777779 MMSSubject=Test

cscript sendmms.vbs +4477777777 image.gif "MMSText=The content of this message
has been forward locked" "MMSSubject=Forward Lock Test" MMSForwardLock=Yes
```

Additional variable options:

MMSWAPPush=Yes - indicates that the message being sent should be sent as an "Multimedia Content Push" message via WAP Push instead of as an MMS message.

MMSSMSPush=Yes - indicates that the message being sent should be sent as an "Multimedia Content Push" message via a URL link in a text SMS message instead of as an MMS message.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Now SMS/MMS Proprietary URL Submission

The Now SMS/MMS Proprietary format for submission of an MMS message is the interface that is used by the "Send MMS Message" form in the web menu interface of the gateway.

To submit a message via this interface, a user account must be specified on the "**SMS Users**" configuration dialog.

To submit it the same way that the "Send MMS Message" form does in the gateway's web menu interface, you need to do an HTTP POST in the "multipart/form-data" MIME type.

Basically, when you POST, it would look something like this:

```
POST / HTTP/1.0
Accept: */*
Content-type: multipart/form-data; boundary="--boundary-border--"
Content-length: xxxxx (size of content part of post)
Authorization: username:password (base64 encoded)

----boundary-border--
Content-Disposition: form-data; name="PhoneNumber"

+448080148324
----boundary-border--
Content-Disposition: form-data; name="MMSFrom"

sender@domain (or +38484753009)
----boundary-border--
Content-Disposition: form-data; name="MMSSubject"

Message Subject
----boundary-border--
Content-Disposition: form-data; name="MMSText"

An optional text part of the message.
----boundary-border--
Content-Disposition: form-data; name="MMSFile"; filename="original-filename.ext"
Content-type: Mime/Type

File data goes here
----boundary-border----
```

The content-type for the overall message is "multipart/form-data". As with other multipart MIME encoding, you must include a boundary that separates the multiple parts of the message.

It is very important to set the Content-length: field to specify the length of the multipart content that follows (this is how the server knows your HTTP post is complete).

The Authorization header specifies the username and password used to login to the gateway. It is in the format "username:password" and is Base64 encoded.

Then, the content has a part for each form variable.

The "PhoneNumber" variable is required, it is the phone number of the message recipient.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The "MMSFrom" variable is optional. It is the "From:" address to be used in the MMS message. (If sending a pre-compiled MMS message, this is used for the notification only.)

The "MMSSubject" variable is optional. It is the "Subject" line used in the MMS message. (If sending a pre-compiled MMS message, this is used for the notification only.)

The "MMSText" variable is optional. It contains a text part of the message.

The "MMSFile" variable is optional, and can be repeated multiple times. It contains binary file content for an uploaded file. If you're sending a pre-compiled MMS file, you'd only include the "MMSFile" variable once. If you're sending a message for the gateway to compile, you could include each of the individual message parts as a separate instance of the "MMSFile" variable.

As an alternative to using the HTTP POST, if the content of the MMS message already exists on a web server, the "MMSFile" variable can be specified as a URL instead of the actual file content. In this case, the message can be submitted to the gateway with an HTTP GET request, instead of requiring HTTP POST. For example:

```
http://127.0.0.1:8800/?
PhoneNumber=xxxxxx&MMSFrom=sender@domain&MMSSubject=Message+Subject&MMSText=An+
optional+text+part+of+the+message&MMSFile=http://www.nowsms.com/media/logo.gif
```

The variables are the same as described above, except that in a GET request, the "MMSFile" variable must point to a URL. As with the POST request, the "MMSFile" variable can be repeated to specify multiple content files.

Note: If authentication is enabled for the web interface, any application submitting a message must supply a user name and password for access. This user name and password refers to an account defined on the "SMS Users" configuration dialog. The application can either include the user name and password in an "Authorization:" header using "HTTP Basic Authentication", or it can include "&User=xxxx&Password=xxxx" parameters within the URL request.

Note: A PHP script that simplifies the process of generating an HTTP POST for submitting MMS messages to NowSMS can be found at http://www.nowsms.com/discus/messages/1/1113.html.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MM7

Before submitting a message via MM7, a VASP (Value Added Service Provider) account must be defined to the Now SMS/MMS Gateway. This account is defined on the "**MMSC VASP**" configuration dialog.

To post to the Now SMS/MMS Gateway via MM7, you must connect to the HTTP port configured for the MMSC on the "**MMSC**" configuration dialog. And you must perform an HTTP POST of the MM7 content to a URI of "/mm7", which is how the gateway knows that the VASP intends to submit in the MM7 format.

Optionally the URI can include the account name and password of the "MMSC VASP" using the format "/mm7/account=password".

The HTTP headers of your POST must include a "Content-length:" header. If the VASP account name is not included in the URI, the request must either include an "Authorization:" header for Basic authentication using the account name and password configured for the account, or it must originate from an IP address that matches the name configured for the VASP account. (If your software cannot generate an "Authorization:" header, it is possible to configure the account name for the VASP as an IP address, and in this case, the MMSC will recognise any connections from that IP address as being for this VASP account.)

The "Content-type:" header in the POST should be one of the "multipart" types (usually "multipart/related"), and should include a "boundary=" parameter that delimits the different parts of the message.

The first part of the multipart message is expected to be the XML for the MM7 request, and we're going to expect to see a <Recipients> section with at least one <To>, <Cc> or <Bcc> recipient specified.

The second part of the multipart message is expected to be the content for the MMS message, and this in turn will usually be another MIME multipart structure.

The following example is adapted from the official MM7 specification that is included in the 3GPP TS 23.140 specification:

Note that this example does not include a SMIL file, and as part of the MMS content, you would probably want to include a SMIL file (application/smil), which this example does not include.

Also note that the MM7 XML portion of the document (the first part of the main multipart content) should not use any Content-Transfer-Encoding, it should always be expressed without any encoding. For the portion of the document that includes the MMS content itself, you can use Content-Transfer-Encoding of either quoted-printable or base64, or no encoding can be specified in which case it is assumed that the binary data is to be included as is.

```
POST /mm7 HTTP/1.1
Host: mms.omms.com
Content-Type: multipart/related; boundary="NextPart_000_0028_01C19839.84698430";
type=text/xml;  start="</tnn-200102/mm7-submit>"
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
Content-Length: nnnn
SOAPAction: ""

--NextPart_000_0028_01C19839.84698430
Content-Type:text/xml; charset="utf-8"
Content-ID: </tnn-200102/mm7-submit>

<?xml version="1.0" ?>
<env:Envelope xmlns:env="http://schemas.xmlsoap.org/soap/envelope/">
<env:Header>
<mm7:TransactionID
xmlns:mm7="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/schema/REL-5-M M7-1-3"
env:mustUnderstand="1">
vas00001-sub
</mm7:TransactionID>
</env:Header>
<env:Body>
<SubmitReq xmlns="http://www.3gpp.org/ftp/Specs/archive/23_series/23.140/schema/REL-5-
MM7-1 -3">
<MM7Version>5.6.0</MM7Version>
<SenderIdentification>
<VASPID>TNN</VASPID>
<VASID>News</VASID>
</SenderIdentification>
<Recipients>
<To>
<Number>7255441234</Number>
</To>
<Cc>
<Number>7255443333</Number>
</Cc>
<Bcc>
<RFC2822Address>7255444444@OMMS.com</RFC2822Address>
</Bcc>
</Recipients>
<ServiceCode>gold-sp33-im42</ServiceCode>
<LinkedID>mms00016666</LinkedID>
<MessageClass>Informational</MessageClass>
<TimeStamp>2002-01-02T09:30:47-05:00</TimeStamp>
<EarliestDeliveryTime>2002-01-02T09:30:47-05:00</EarliestDeliveryTime>
<ExpiryDate>P90D</ExpiryDate>
<DeliveryReport>true</DeliveryReport>
<Priority>Normal</Priority>
<Subject>News for today</Subject>
<ChargedParty>Sender</ChargedParty>
<DistributionIndicator>true</DistributionIndicator>
<Content href="cid:SaturnPics-01020930@news.tnn.com" allowAdaptations="true"/>
</SubmitReq>
</env:Body>
</env:Envelope>

--NextPart_000_0028_01C19839.84698430
Content-Type: multipart/mixed; boundary="StoryParts-74526-8432-2002-77645"
Content-ID:<SaturnPics-01020930@news.tnn.com>


--StoryParts-74526-8432-2002-77645
Content-Type: text/plain; charset="us-ascii"

Science news, new Saturn pictures...

--StoryParts-74526-8432-2002-77645
Content-Type: image/gif
Content-ID:<saturn.gif>
Content-Transfer-Encoding: base64

R0lGODdhZAAwAOMAAAAAAIGJjGltcDE0OOfWo6Ochbi1n1pmcbGojpKbnP/lpW54fBMTE1RYXEFO

...

--StoryParts 74526-8432-2002-77645--
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
--NextPart_000_0028_01C19839.84698430--
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MM4

Before submitting a message via MM4, a VASP (Value Added Service Provider) account must be defined to the Now SMS/MMS Gateway. This account is defined on the "**MMSC VASP**" configuration dialog.

To post to the NowSMS MMSC via MM4, your application is making an SMTP connection to the MMSC, and you would connect to the "SMTP Port Number" that is configured on the "MMSC" configuration dialog.

To submit a message via MM4, the VASP must authenticate via SMTP using the "AUTH LOGIN" approach, or similar to the way it works for MM7, the VASP account can be created with an IP address as the "Account Name", and in that case any connection from that IP will be accepted as being a connection from that VASP account.

"AUTH LOGIN" is rather simple. Basically, as part of the SMTP dialog, after the initial HELO or EHLO command, the SMTP client needs to issue the command AUTH LOGIN. The SMTP server responds with a "300" series code and prompts for the account name (the prompt after the code is a BASE64 encoded string). The client sends the account name as a BASE64 encoded string. The SMTP server responds with another "300" series code prompting for the password, and the client responds with the password as a BASE64 encoded string. A "200" series response indicates that the authentication was accepted, a "500" series response indicates that it was not.

The SMTP dialog then continues as normal, generally with a "MAIL FROM:" command from the client indicating the sending address of the message, followed by one or more "RCPT TO:" commands to indicate the recipients for the message. Note that NowSMS expects the "RCPT TO:" addresses to be in a format of phonenumber@domain.name or phonenumber/TYPE=PLMN@domain.name, where "domain.name" is the "Domain Name for MMS E-Mail" configured on the "MMSC" configuration dialog. (The "Local Host Name or IP Address" value is also acceptable here.) If the domain name is not present, the MMSC will reject the recipient. As special support for Multimedia WAP Push, the following address formats are also supported to specify Multimedia WAP Push to be used for sending to the phone number: phonenumber.wappush@domain.name or phonenumber/TYPE=WAPP@domain.name.

The actual MMS message is then transmitted via the "DATA" command. Normally an MMS message would be a multipart MIME message with multiple content parts, although NowSMS will also accept a message that includes only a single part.

The MM4/SMTP dialog looks something like this:

The dialog looks something like this (IN means from server, OUT means from client):

```
IN: 220 SMTP Ready
OUT: HELO client.name (or EHLO client.name)
IN: 250 OK (or a multiline response if EHLO was used)
OUT: AUTH LOGIN
IN: 334 VXNlcm5hbWU6
("Username:" BASE64 encoded)
OUT: dGVzdA==
("test" BASE64 encoded)
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
IN: 334 UGFzc3dvcmQ6
("Password:" BASE64 encoded)
OUT: dGVzdA==
("test" BASE64 encoded)
IN: 235 Ok
OUT: MAIL FROM: <username@domain.com>
IN: 250 Ok
OUT: RCPT TO: <+447778889999/TYPE=PLMN@mms.domain.com>
IN: 250 Ok
OUT: DATA
IN: 354 Ok, end with "." on a new line...
OUT: (Transmit MIME encoded message, then end with a line with only the . character)
IN: 250 Message Accepted
OUT: QUIT
```

Using the example data from the MM7 message above, the MIME encoded message would look something like this:

```
To: +447778889999/TYPE=PLMN@mms.domain.com
From: username@domain.com
Subject: News for today
Content-Type: multipart/mixed; boundary="StoryParts-74526-8432-2002-77645"
Content-ID:<SaturnPics-01020930@news.tnn.com>

--StoryParts-74526-8432-2002-77645
Content-Type: text/plain; charset="us-ascii"

Science news, new Saturn pictures...

--StoryParts-74526-8432-2002-77645
Content-Type: image/gif
Content-ID:<saturn.gif>
Content-Transfer-Encoding: base64

R0lGODdhZAAwAOMAAAAAAIGJjGltcDE0OOfWo6Ochbi1n1pmcbGojpKbnP/lpW54fBMTE1RYXEFO

...

--StoryParts-74526-8432-2002-77645--
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MM1

Before submitting a message via MM1, a VASP (Value Added Service Provider) account must be defined to the Now SMS/MMS Gateway. This account is defined on the "**MMSC VASP**" configuration dialog.

To post to the Now SMS/MMS Gateway via MM1, you must connect to the HTTP port configured for the MMSC on the "MMSC" configuration dialog. And you must perform an HTTP POST of the MM1 content to a URI of "/mm1", which is how the gateway knows that the VASP intends to submit in the MM1 format.

The HTTP headers of your POST must include a "Content-length:" header. To specify the account name and password of the VASP account, you must either include an "Authorization:" header for Basic authentication using the account name and password configured for the VASP account, or the account name and password can be specified on the request URI (e.g., /mm1/account=password). Alternatively, the request must originate from an IP address that matches the name configured for the VASP account. (If your software cannot generate an "Authorization:" header, it is possible to configure the account name for the VASP as an IP address, and in this case, the MMSC will recognise any connections from that IP address as being for this VASP account.)

The "Content-type:" header in the POST should be "application/vnd.wap.mms.message", and the message should be a binary MMS message of the m-send-req format, formatted according to the MMS Encapsulation Specification, published by the Open Mobile Alliance.

Consistent with HTTP specifications, a "400" or "500" series HTTP response would be considered an error condition. The MMSC might return a "200" series response even if an error did occur, in which case information would be included in the "X-Mms-response-status" field of the MM1 response (MIME type "application/vnd.wap.mms-message", X-Mms-Message-Type of "m-send-conf") would contain a response-status value other than Ok.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

EAIF

Before submitting a message via EAIF, a VASP (Value Added Service Provider) account must be defined to the Now SMS/MMS Gateway. This account is defined on the "**MMSC VASP**" configuration dialog.

To post to the Now SMS/MMS Gateway via EAIF, you must connect to the HTTP port configured for the MMSC on the "MMSC" configuration dialog. And you must perform an HTTP POST of the EAIF content to a URI of "/eaif", which is how the gateway knows that the VASP intends to submit in the EAIF format.

The HTTP headers of your POST must include a "Content-length:" header. To specify the account name and password of the VASP account, you must either include an "Authorization:" header for Basic authentication using the account name and password configured for the VASP account, or the account name and password can be specified on the request URI (e.g., /eaif/account=password). Alternatively, the request must originate from an IP address that matches the name configured for the VASP account. (If your software cannot generate an "Authorization:" header, it is possible to configure the account name for the VASP as an IP address, and in this case, the MMSC will recognise any connections from that IP address as being for this VASP account.)

The "Content-type:" header in the POST should be "application/vnd.wap.mms.message", and the message should be a binary MMS message of the m-send-req format, formatted according to the MMS Encapsulation Specification, published by the Open Mobile Alliance.

MMS message recipients can be specified either in the MMS message content itself, or in the "X-Nokia-MMSC-To:" header.

Consistent with HTTP and EAIF specifications, a "400" or "500" series HTTP response would be considered an error condition. The expected response for a valid message submission would be an HTTP "204" response that includes an "X-Nokia-MMSC-Message-Id:" header. (In beta releases of the v5.0 Now SMS/MMS Gateway, the MMSC might return a "200" series response even if an error did occur, in which case information would be included in the "X-Mms-response-status" field of an MM1 response with the "X-Mms-Message-Type" field containing a response-status value other than Ok. We expect this to have been corrected prior to the v5.0 release.)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Submitting SMS Messages - URL Parameters

To send an SMS message via a menu driven interface, please see the help section titled "**Web Menu Interface**" on page 73. This section describes how to send a text message programmatically via URL parameters.

To send a message via SMS, use the following URL format:

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&...
```

For 127.0.0.1, please substitute the IP address or host name assigned to your gateway PC. (Note: 127.0.0.1 is a local loop back address that can be utilized when you are connecting to the gateway from the same computer.)

For 8800, please substitute the port number that the gateway is configured to use.

Substitute the phone number that you wish to send the SMS message to for the "xxxxxxxx" in the "PhoneNumber" parameter. Use either the local phone number format, or the international phone number format (your network provider may or may not allow you to send to international phone numbers). If the international phone number format is used, note that you must substitute "%2B" for the "+" character, because of URL escaping restrictions. For example, to send an SMS to +447778001210, use the following URL format:

```
http://127.0.0.1:8800/?PhoneNumber=%2B447778001210&...
```

Parameters after the "..." are dependent on the type of message being sent. The following table summarizes available URL parameters:

URL Parameter	Message Type	Description
PhoneNumber	All	Recipient phone number or distribution list name. This field can contain a comma delimited list of recipient phone numbers and/or distribution list names.
User	All	If authentication is enabled for the web interface, any application submitting a message must supply a user name and password for access. This user name and password refers to an account defined on the "SMS Users" configuration dialog. The application can either include the user name and password in an "Authorization:" header using "HTTP Basic Authentication", or it can include the "User" and "Password"

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

		parameters in the URL request.
Password	All	See "User" parameter above.
Text	Text SMS, Binary SMS, WAP Push, WAP OTA Bookmark	For SMS text messages, this specifies the text of the message. For binary SMS messages, this is a string of hexadecimal characters representing the data being sent in the binary message. For WAP Push messages, this is the text associated with a Service Indication (SI) Push. For a WAP OTA Bookmark, this is the text name of the bookmark.
Data	Binary SMS	Interchangeable with the "Text" parameter. Officially documented only for binary SMS messages.
UDH	Binary SMS	A text string of hexadecimal characters representing the User Data Header (UDH) of the binary SMS message.
DCS	Text or Binary SMS (limited usage for text SMS)	A hex value representing the value of the SMS Data Coding Scheme (DCS) for this message. F5 is a common value for most binary SMS message types in GSM environments. Another common DCS setting is 10 for sending a flash (Class 0) message *(see page 75)*.
PID	Text or Binary SMS	A hex value representing the SMS Protocol ID (PID) of this SMS message. The most frequent use of the PID parameter is for sending replacement type messages *(see page 75)*.
Binary	Binary SMS	Set to 1 for binary message submission
Sender	All	Sender phone number for this SMS message.
Charset	Text SMS	Specifies the character set used for the text in the "Text" parameter. By default, NowSMS assumes UTF-8. NowSMS supports any of the character sets supported by Windows. Common settings include "iso-8859-1" for Western Europe, "iso-8859-6" for Arabic, and "big5" for Chinese.
DelayUntil	Text or Binary SMS	This parameter allows messages to be submitted to NowSMS and queued for later processing. The value of this parameter should be of the format "YYYYMMDDHHMM", indicating the date and time until which the message should be delayed, where YYYY is the year, MM

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

		is the month, DD is the day, HH is the hour (in 24 hour format), and MM is the minutes.
ReceiptRequested	All	Set to "Yes" if a delivery or non-delivery receipt is requested for this message. (Not supported by all SMSC interfaces.)
ReplyRequested	Text SMS	***Supported by GSM Modem and SMPP SMSC interfaces only***. Set this parameter to "Yes" to set the reply path flag. Technically this indicates that you are requesting that the receiving user be able to send a reply back via the same SMSC through which this message is being submitted. In practice, some users use this setting because some phones will display a prompt on the receiving device indicating that the sender is requesting a reply. *(Note that some SMSCs will reject messages sent with this flag, and others may ignore the setting.)*
VoiceMail	Voice Mail Notification	"On" - Turn on voice mail waiting indication "Off" - Turn off voice mail waiting indication "FaxOn" - Turn on fax message waiting indicator "FaxOff" - Turn off fax message waiting indicator "EmailOn" - Turn on e-mail message waiting indicator "EmailOff" - Turn off e-mail message waiting indicator "VideoOn" - Turn on video message waiting indicator "VideoOff" - Turn off video message waiting indicator "OtherOn" - Turn on other message waiting indicator "OtherOff" - Turn off other message waiting indicator
VoiceMailMessageCount	Voice Mail Notification	Specifies an optional "message count" for the number of messages waiting associated with this notification.
WAPURL	WAP Push	URL to be sent in the WAP Push message.
WAPPushInitiatorURI	WAP Push, OMA OTA	Sets the WAP Push Initiator URI. For more information, refer to the Technical Bulletin titled **WAP Push or OTA: Unknown Sender**.
WAPPushFlag	WAP Push,	Sets the WAP Push Flag. For more

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

	OMA OTA	information, refer to the Technical Bulletin titled **WAP Push or OTA: Unknown Sender**.
WAPSIID	WAP Push (Service Indication)	"Service Indication ID" is a text string that defines an id string to be associated with a service indication push. If a push has an "WAPSIID" associated with it, it is possible to later send a "WAPSIAction=delete" push with the same "WAPSIID" value to delete the previous push message from the device inbox. Similarly, if a mobile device receives a push message with a "WAPSIID" that matches that of a previously received push that is still in its inbox, the new push message should replace the existing push message.
WAPSIACTION	WAP Push (Service Indication)	"Signal Action" specifies the type of alert to be associated with the push. While this is not very widely supported, the general intent is to associate a priority with the alert. Valid settings are: "signal-high" - high priority alert "signal-medium" - medium priority alert "signal-low" - low priority alert "signal-none" - do not generate a notification alert for this push "delete" - if a previously sent push exists in the device inbox, with the same WAPSIID as a specified in this push, then the push should be deleted from the device inbox.
WAPSIEXPIRES	WAP Push (Service Indication)	The "SI Expires" field specifies a date/time at which the receiving device should automatically expire the push. This is a date/time value relative to GMT, in the format "yyyy-mm-ddThh:mm:ssZ". For example, "2006-02-24T00:00:00Z".
WAPSICREATED	WAP Push (Service Indication)	The "SI Created" field specifies a creation date/time stamp to be associated with the push. If specified, this date/time stamp should take the format "yyyy-mm-ddThh:mm:ssZ", specifying a date/time value relative to GMT.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

		For example, "2006-02-24T00:00:00Z".
WAPSL	WAP Push (Service Load)	When the "WAPURL" parameter is specified, set this parameter to any value to send the WAP Push as a "Service Load" (SL) message, instead of the default "Service Indication" (SI) message.
WAPSLAction	WAP Push (Service Load)	Specifies the type of action to be taken upon receipt of a "Service Load" push. Valid settings are: "execute-low" - The browser fetches the URL and executes it in a non-intrusive manner "execute-high" - The browser fetches the URL, executes it and displays it in a manner that may be considered intrusive "cache" - The browser fetches the URL and saves the resulting data in the browser's cache *(if a cache does not exist, the push is ignored)*
MMSText	MMS Message, Multimedia WAP Push	Text to be included when sending an MMS Message.
MMSFile	MMS Message, Multimedia WAP Push	Contains the contents of an uploaded file posted via a form with a MIME encoding of "multipart/form-data", or specifies a HTTP URL pointing to the file content when specified in a GET request. This parameter can be repeated multiple times to indicate multiple files to be included in the content of the MMS message.
MMSSubject	MMS Message, MMS Notification, Multimedia WAP Push	Subject for the MMS Message or MMS Notification Message.
MMSFrom	MMS Message, MMS Notification, Multimedia WAP Push	Sender for the MMS Message or MMS Notification Message
MMSDeliveryReport	MMS Message	"Delivery Report specifies whether or not a delivery report is requested for the message. Set to "Yes" to request a delivery report. Note that any delivery report would be directed back to the phone number or e-mail address specified in the "MMSFrom" address.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSReadReport	MMS Message	"Read Report" specifies whether or not a read receipt is requested for the message. Note that the receiving client may choose not to send a read receipt. Any read receipt report would be directed back to the phone number or e-mail address specified in the "MMSFrom" address.
MMSPriority	MMS Message	"Priority" is a user defined priority to be associated with the message. Generally, any priority definition associated with the message is ignored by the underlying transport, but the receiving client may decide to display messages differently based upon this priority setting.
MMSMessageClass	MMS Message	"Message Class" is an attribute defined in the MMS specifications. "Personal" is the message type that is used for standard user-to-user communications. Other defined message classes that are supported by this parameter include: "Informational" "Advertisement"
MMSForwardLock	MMS Message, Multimedia WAP Push	Forward locking is the most basic level of DRM (Digital Rights Management). When "Forward Lock" is set to "Yes", this indicates that the receiving device should not allow any non-text objects in the message to be forwarded off of the device. The device may allow the user to extract pictures, videos or sounds from the message and save them on the phone. However, any such objects remain forward locked, such that they cannot be forwarded to another user or transferred to another device.
DRMRestrict	MMS Message, Multimedia WAP Push	Beyond forward locking, More advanced DRM (Digital Rights Management) restrictions can be applied to limit the number of times that the user can access an object, or start and end dates can be specified to limit how long the user can access an object. These advanced DRM restrictions can be applied by setting "DRMRestrict" to "Yes". When this setting is enabled, forward lock is also implied,

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

		and the value of the "MMSForwardLock" setting is ignored.
DRMRestrictTextXML	MMS Message, Multimedia WAP Push	"Yes" specifies that the rights object should be encoded in text XML format. "No" specfies that the rights object should be encoded in binary XML format. The default is "No".
		"DRM Permissions" specify what types of access are allowed against the objects in a message that is protected with DRM. For example, an audio or video object requires "Play" permission before the user can access it. An image requires "Display" permission before the user can access it, and it requires "Print" permission if the user is to be allowed to print the image to a printer , perhaps over Bluetooth. An application requires "Execute" permission before the user can make use of the application. In all cases, the forward locking is assumed, so that the user is not allowed to forward or transfer the object from the device. If you are sending multiple types of objects in the MMS message, check all permissions that are required for the different types of objects that are being sent.
DRMPermissionPlay	MMS Message, Multimedia WAP Push	Set to "Yes" to enable DRM "Play" Permission.
DRMPermissionDisplay	MMS Message, Multimedia WAP Push	Set to "Yes" to enable DRM "Display" Permission.
DRMPermissionExecute	MMS Message, Multimedia WAP Push	Set to "Yes" to enable DRM "Execute" Permission.
DRMPermissionPrint	MMS Message, Multimedia WAP Push	Set to "Yes" to enable DRM "Print" Permission.
		"DRM Constraints" specify constraints with regard to how long a DRM protected object object should remain accessible to the user. It is possible to specify one or more of these constraints that follow.
DRMConstraintCount	MMS Message,	"# of Accesses (count)" specifies the

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

	Multimedia WAP Push	the user can only access the DRM protected object this number of times before access is no longer allowed.
DRMConstraintStart	MMS Message, Multimedia WAP Push	"Start Date (yyyy-mm-dd)" specifies that the user will not be allowed to access the DRM protected object until on or after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2006-02-24.)
DRMConstraintEnd	MMS Message, Multimedia WAP Push	"End Date (yyyy-mm-dd)" specifies that the user will not be allowed to access the DRM protected object after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2006-02-24.)
DRMConstraintInterval	MMS Message, Multimedia WAP Push	"# of Days (interval)" specifies that the user will be allowed to access the DRM protected object for this number of days after initial receipt of the object. The user can either enter a number of days here, or they can enter any valid value defined for the "<interval>" element in the OMA DRM Rights Expression Language specification. For example, P2Y10M15DT10H30M20S represents a duration of 2 years, 10 months, 15 days, 10 hours, 30 minutes and 20 seconds.
MMSWAPPush	Multimedia WAP Push	Set to "Yes" to indicate that the message being sent should be sent as an "Multimedia WAP Push" message instead of as an MMS message.
MMWAPTemplate	Multimedia WAP Push	Specifies a WML template to be used for generating the Multimedia WAP Push message. Please refer to the NowSMS discussion board for more information.
MMWAPURLOnly	Multimedia WAP Push	If set to "Yes", specifies that NowSMS should return a URL pointer for a dynamically generated "Multimedia WAP Push" in the HTTP response, however NowSMS does not generate the actual WAP Push message for sending the content. Please refer to the NowSMS discussion board for more information.
MMSURL	MMS Notification	URL that contains the MMS Message content for which an MMS Notification Message should be generated.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSURLValidate	MMS Notification	Normally, when NowSMS generates an MMS Notification, it first validates that the specified "MMSURL" can be retrieved, and that it is of a valid format. Include this parameter, set to "No" to disable this validation check.
WAPBookmark	WAP Bookmark	URL to be sent as a WAP OTA bookmark (not supported by many devices).
OTA	WAP OTA Config	Name of an ".ota" file in the OTA subdirectory which contains WAP OTA configuration information, or value "POST" when OTA content is being submitted via HTTP POST.
OTAOMA	OMA Provisioning Content OTA	Name of an ".ota" file in the OTA subdirectory which contains an OMA Provisioning Content document, or value "POST" when provisioning content is being submitted via HTTP POST.
OTAPINTYPE	OMA Provisioning Content OTA	Specifies the type of PIN specified in the OTAPIN variable. Can either be a value of USERPIN, NETWPIN, or USERNETWPIN.
OTAPIN	OMA Provisioning Content OTA	The value of this parameter depends upon the value of the OTAPINTYPE parameter. USERPIN – The PIN a short PIN code (often 4 digits). When the user receives the OTA settings message, they will need to supply this PIN code in order to be able to open the message and apply the settings. NETWPIN – The PIN is a network PIN code. In the GSM environment, this is the IMSI number associated with the SIM card in the device. (Hint, if you want to experiment with determining the PIN card associated with a SIM, you can put the SIM into a GSM modem and the AT+CIMI command to return the IMSI. However, not all GSM modems support the AT+CIMI command.) When the device receives the settings, if the NETWPIN does not match the IMSI, the settings will be discarded. USERNETWPIN – The PIN is a

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

		combination of the USERPIN and NETWPIN types. Define the OTA PIN as the IMSI number associated with the SIM card in the device, followed by a ":" character, followed by a USERPIN *(e.g., 1234567889012345:1234)*. When the device receives the settings, the user will be prompted for a PIN. This user supplied PIN, and the SIM card IMSI, must match in order for the settings to be accepted.
Validity	All	***Supported by outbound GSM modem and SMPP SMSC connections only***. This parameter specifies a validity period for the message as an interval defined in hours, minutes or days. If the SMSC cannot deliver the message within the specified validity period, the SMSC is instructed to discard the message. *(Note that this setting is not supported by all SMSCs.)* Specify ##D for a validity period in days, ##H for a validity period in hours, or ##M for a validity period in minutes, where ## is a numeric value *(e.g., 30D for 30 days or 7H for 7 hours)*.
ContinueURL	All	URL to continue to after SMS message submission.
SourcePort	All	Specifies the source port number as a hex string, that is associated with the sender address of this message. (Used for application/port addressing of messages to Java MIDlets.)
DestPort	All	Specifies the source port number as a hex string, that is associated with the recipient(s) of this message. (Used for application/port addressing of messages to Java MIDlets.)
BillingInfo	All	***Supported by outbound UCP/EMI SMSC connections only***. This parameter specifies a value to be included as the "billing info" parameter in the outbound SMS message, when it is sent via a UCP/EMI connection.
ServiceType	All	Supported by outbound SMPP SMSC connections. This parameter specifies a value for the "service_type" parameter in the outbound SMS message when it is sent via an SMPP connection.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

LocalUser	All	Indicates that the message should be routed to a local "SMS Users" account supplied as the parameter value. The account must have SMPP or SMTP Login enabled to support receiving messages.
InboundMessage	All	Set to "Yes" to indicate that this is an inbound/received message that should be routed to the 2-way command processor instead of being routed to an outbound SMSC connection.
SMSCRoute	All	Specifies the name of an SMSC through which this message should be routed *(e.g., "Bluetooth Modem" or "SMPP - a.b.c.d:xyz")*. Or, instead of using a specific SMSC name, it can be a route name that is defined as associated with one or more SMSCs. To define a route name for an SMSC, it is necessary to manually edit SMSGW.INI, and under the appropriate section header (e.g., [Modem - Bluetooth Modem] or [SMPP - a.b.c.d:xyz]), add RouteName=xxxxx. It is possible for multiple SMSCs to share the same route name, meaning that if a message is submitted via HTTP with the "&SMSCRoute=xxxxx" parameter, it will be routed outbound over the first available SMSC that is configured with the RouteName=xxxxx setting. For more information, see **SMS Message Routing Logic** on page 54.
EMSText	EMS Text	The "EMSText" parameter defines an EMS text message to be sent. This text can include EMS attributes for text formatting, such as bold, italics and large or small text. EMS text messages can also included predefined animations and sounds which are pre-loaded on EMS compatible phones. NowSMS also supports generating the EMS text formatting codes to specify colors to be used in text messages, however this functionality does not appear to be very widely supported in current handsets. NowSMS implements

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

		special tags to indicate EMS attributes in the "EMSText" parameter. For more information on these tags, please see **Sending EMS Messages** on page 205.
RingToneDataText	EMS Ringtone	A text string that contains ring tone data in either RTTTL, iMelody or MIDI format. *(MIDI is a binary format, therefore MIDI data must be represented as a hex string when using this parameter.)*
RingToneDataFile	EMS Ringtone	Ring tone data submitting using HTTP file upload. The file can contain ring tone data in either RTTTL, iMelody or MIDI format. *(Note: This parameter can only be used in an HTTP POST with the content type of multipart/form-data.)*
RingToneDataURL	EMS Ringtone	HTTP URL pointer to a ring tone file residing on another web server. The ring tone file can be in either RTTTL, iMelody or MIDI format.
RingToneOut	EMS Ringtone	Specifies the output format to be used for the ring tone: "Nokia" (Nokia Smart Messaging) - This is the binary encoding for RTTTL, which was originally defined by Nokia. (Note that NowSMS currently only supports the sending these messages out in binary format. The text "//SCKL" format may be supported in a future release.) "EMS" (EMS - iMelody) – The ring tone is converted to iMelody, if necessary, and encoded as an EMS message. "EMSShort" (EMS Short Format - iMelody without headers) – The ring tone is converted to iMelody, if necessary. As EMS can be rather verbose, the headers are stripped from the iMelody data, and it is then encoded as an EMS message. "WAPPush" (WAP Push - MIDI or no conversion) – If the input ring tone is in RTTTL or iMelody format, it is converted to MIDI. Otherwise, no conversion is performed. The output ring tone is delivered as a WAP Multimedia Message.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

PictureMessageDataText	EMS Picture Message	A text string that contains image data in either BMP, JPEG or GIF format. *(These are all binary formats, therefore the image data must be represented as a hex string when using this parameter.)*
PictureMessageDataFile	EMS Picture Message	Image data submitting using HTTP file upload. The file can contain image data in either BMP, JPEG or GIF format. *(Note: This parameter can only be used in an HTTP POST with the content type of multipart/form-data.)*
PictureMessageDataURL	EMS Picture Message	HTTP URL pointer to an image file residing on another web server. The image file can be in either BMP, JPEG or GIF format.
PictureMessageOut	EMS Picture Message	Specifies the output format to be used for the picture message: "Nokia" (Nokia Smart Messaging) - This binary encoding format was originally defined by Nokia. (Note that NowSMS currently only supports the sending these messages out in binary format. The text "//SCKL" format may be supported in a future release.) "EMS" – The image is converted to EMS format, if necessary, and encoded as an EMS message. "WAPPush" (WAP Push - no conversion) – No conversion is performed on the image, and it is delivered as a WAP Multimedia Message.
SMPPOption_*	All	See SMSGW.INI, [SMPPOptions] on page 342.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

When a text message is being submitted via the "Text" parameter, note that due to URL escaping restrictions, space characters should be replaced with "+" characters. Also, certain special characters, such as "?", "&", ":" and "=" need to be replaced with an escape character. The gateway expects characters to be encoded in UTF-8 (Unicode-based) format, therefore some characters, including the Euro (€) may require multiple escaped characters. (Note: The Web Menu Interface automatically performs this escaping.) The following table shows common characters that must be escaped:

"	%22
<	%3C
>	%3E
&	%26
+	%2B
#	%23
%	%25
*	%2A
!	%21
,	%2C
'	%27
\	%5C
=	%3D
€	%E2%82%AC

Message text up to 160 characters in length can be sent in a single SMS message. The gateway automatically supports the sending of longer messages by utilizing "concatenated SMS" to send messages larger than 160 characters in length. Note that some older mobile phones will not support longer SMS messages. For longer SMS messages, one message is sent for every 153 characters of text in the message.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending Text Messages

To send a text SMS message via a menu driven interface, please see the help section titled **Web Menu Interface** on page 73. This section describes how to send a text message programmatically via URL parameters.

To send a text message via SMS, use the following URL format:

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&Text=abc+def+ghi
```

For 127.0.0.1, please substitute the IP address or host name assigned to your gateway PC. (Note: 127.0.0.1 is a local loop back address that can be utilized when you are connecting to the gateway from the same computer.)

For 8800, please substitute the port number that the gateway is configured to use.

Substitute the phone number that you wish to send the SMS message to for the "xxxxxxxx" in the "PhoneNumber" parameter. Use either the local phone number format, or the international phone number format (your network provider may or may not allow you to send to international phone numbers). If the international phone number format is used, note that you must substitute "%2B" for the "+" character, because of URL escaping restrictions. For example, to send an SMS to +447778001210, use the following URL format:

```
http://127.0.0.1:8800/?PhoneNumber=%2B447778001210&Text=abc+def+ghi
```

Substitute the text of the SMS message in the "Text" parameter. Note that due to URL escaping restrictions, space characters should be replaced with "+" characters. Also, certain special characters, such as "?", "&", ":" and "=" need to be replaced with an escape character. The gateway expects characters to be encoded in UTF-8 (Unicode-based) format, therefore some characters, including the Euro (€) may require multiple escaped characters. (Note: The Web Menu Interface automatically performs this escaping.) The following table shows common characters that must be escaped:

"	%22
<	%3C
>	%3E
&	%26
+	%2B
#	%23
%	%25
*	%2A
!	%21
,	%2C
'	%27
\	%5C
=	%3D
€	%E2%82%AC

It is possible to use the "&charset=xxxxxxx" parameter to indicate that the text has been encoded using a character set other than UTF-8. NowSMS supports any of the character

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

sets supported by Windows. Common "&charset=xxxxxxxx" settings include "iso-8859-1" for Western Europe, "iso-8859-6" for Arabic, and "big5" for Chinese

Message text up to 160 characters in length can be sent in a single SMS message. The gateway automatically supports the sending of longer messages by utilizing "concatenated SMS" to send messages larger than 160 characters in length. Note that some older mobile phones will not support longer SMS messages. For longer SMS messages, one message is sent for every 153 characters of text in the message.

If a text message contains text that is outside of the GSM character set, then it is necessary for the message to be sent out using Unicode encoding. When a message is sent with Unicode encoding, only 70 characters can fit into a single SMS message. If a message that contains Unicode characters is longer than 70 characters, one message is sent for every 63 characters of text in the message.

The following table details the characters that are part of the standard GSM character set:

Hex	0x	1x	2x	3x	4x	5x	6x	7x
x0	@	Δ	SP	0	¡	P	¿	p
x1	£	_	!	1	A	Q	a	q
x2	$	Φ	"	2	B	R	b	r
x3	¥	Γ	#	3	C	S	c	s
x4	è	Λ	¤	4	D	T	d	t
x5	é	Ω	%	5	E	U	e	u
x6	ù	Π	&	6	F	V	f	v
x7	ì	Ψ	'	7	G	W	g	w
x8	ò	Σ	(	8	H	X	h	x
x9	Ç	Θ	)	9	I	Y	i	y
xA	LF	Ξ	*	:	J	Z	j	z
xB	Ø	1)	+	;	K	Ä	k	ä
xC	ø	Æ	,	<	L	Ö	l	ö
xD	CR	æ	-	=	M	Ñ	m	ñ
xE	Å	ß	.	>	N	Ü	n	ü
xF	å	É	/	?	O	§	o	à

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Additionally, there are some characters, such as the Euro (€) which are part of an extended GSM character set, which is detailed in the following table:

Hex	0x	1x	2x	3x	4x	5x	6x	7x
x0					|			
x1								
x2								
x3								
x4		^						
x5							€	
x6								
x7								
x8			}					
x9			{					
xA								
xB								
xC				[				
xD				~				
xE				]				
xF			\					

If a text message contains only characters that are part of either the standard or extended GSM character set table, then it is subject to the 160 character limit (with the exception that any characters in the extended GSM character set are encoded as two characters instead of one). Otherwise, if the text message contains any characters outside of this character set, the entire message must be encoded in Unicode format, subject to the 70 character limit.

Additional supported URL parameters when sending text messages include:

"Sender" – Specifies the phone number to be included as the sender of the message. (Note: Depending on the configuration of the gateway and the SMSCs to which the gateway connects, this value may be ignored.)

"PID" - Specifies a protocol id field to be associated with the message. The web interface of NowSMS includes checkbox settings for specifying a "Message Type" value. Setting one of the "Replacement Type" values means that if the gateway sends a subsequent message with the same replacement type value, this will replace any previous messages that were sent by the same sender with the same replacement type value. When submitting an SMS message via URL parameters, replacement type values 1 thru 7 correspond to settings of PID=41 thru PID=47.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"DCS" - Specifies the data coding scheme value associated with the message. NowSMS normally sets this value as appropriate *(0 indicates a normal text message, and 8 indicates that the message contains Unicode characters)*. The web interface of NowSMS includes checkbox settings for specifying a "Message Class" value. "Message Class" settings are generally used only for testing, except for "Class 0 (Flash)" messages which can be occasionally useful. A "flash" message is an SMS message that is automatically opened on the display of the receiving phone, and is normally not saved to the phone's inbox, so that once the user exits the message, the message automatically disappears.

When submitting an SMS message via URL parameters, message class settings of 0 thru 3 correspond to settings of DCS=10 thru DCS=13. Note that NowSMS will automatically convert these DCS values if the message text contains characters that must be encoded using Unicode characters. However, some SMSC connections, such as SMPP, will not support flash messages that contain Unicode text.

For a complete list of URL parameters, please refer to the section **Submitting SMS Messages - URL Parameters** on page 187 .

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending EMS Messages

The Send EMS Message form contains some options to send some common types of EMS and Nokia Smart Messaging messages.

"**EMS Text**" support allows you to send text messages that include EMS attributes for text formatting, such as bold, italics and large or small text. EMS text messages can also included predefined animations and sounds which are pre-loaded on EMS compatible phones. NowSMS also supports generating the EMS text formatting codes to specify colors to be used in text messages, however this functionality does not appear to be very widely supported in current handsets. EMS Text messages are an interesting messaging option, because the EMS standard is designed in such a way that an EMS text message is gracefully downgraded when delivered to a recipient handset that does not support EMS. If the recipient handset does not understand EMS, it will display only the text of the message.

"**EMS ring tone**" support allows you to submit ring tone data in either RTTTL, iMelody or MIDI format, and send the ring tone out in either EMS, Nokia Smart Messaging, or WAP Push/MIDI format. These formats are largely supported for the sake of interfacing with older or less capable handsets, as newer handsets often support true tone formats based upon MP3.

"**EMS Picture Message**" support allows you to send simple picture messages using either the EMS or Nokia Smart Messaging format. These formats refer to the old monochrome/black & white images that are sent via SMS, as opposed to the more advanced functionality offered by MMS. While there is little use for monochrome images in a Technicolor world, these message types can be useful for applications that need to send bar code types of images via SMS, where they cannot assume that recipient devices will have MMS or WAP capabilities.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending EMS Messages – EMS Text

The "**EMS Text**" facility allows you to send text messages that include EMS attributes for text formatting, such as bold, italics and large or small text. EMS text messages can also included predefined animations and sounds which are pre-loaded on EMS compatible phones. NowSMS also supports generating the EMS text formatting codes to specify colors to be used in text messages, however this functionality does not appear to be very widely supported in current handsets. EMS Text messages are an interesting messaging option, because the EMS standard is designed in such a way that an EMS text message is gracefully downgraded when delivered to a recipient handset that does not support EMS. If the recipient handset does not understand EMS, it will display only the text of the message.

The NowSMS web form includes a simple editor that inserts tags into the message which specify where text attributes should be changed, or where pre-defined animations should be inserted. It may be helpful to experiment with this editor in order to better understand how NowSMS implements these tags.

To send an EMS text message, use the following URL format:

```
http://127.0.0.1:8800/?
PhoneNumber=xxxxxxxx&EMSText=abc+<b>def</b>+ghi
```

or

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&EMSText=abc+%3Cb%3Edef
%3C/b%3E+ghi
```

For 127.0.0.1, please substitute the IP address or host name assigned to your gateway PC. (Note: 127.0.0.1 is a local loop back address that can be utilized when you are connecting to the gateway from the same computer.)

For 8800, please substitute the port number that the gateway is configured to use.

Substitute the phone number that you wish to send the SMS message to for the "xxxxxxxx" in the "PhoneNumber" parameter.

The "EMSText" parameter contains the text of the message to be sent. The format of this parameter is similar to the "Text" parameter that is used when sending a text message. However, the "EMSText" parameter can also include tags that specify where text attributes should be changed, or where pre-defined animations should be inserted.

Note that due to URL escaping restrictions, some characters must be escaped when they appear in a URL. And depending on your programming environment, you may need to escape the "<" and ">" characters that are used in NowSMS EMS attribute tags as "%3C" and "%3E" respectively. For more information on URL escaping, please see **Sending Text Messages** on page 201.

To turn on and off the **bold** text attribute, insert in the text to indicate the beginning of a bold section, and to indicate the end.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

To turn on and off the *italic* text attribute, insert in the text to indicate the beginning of an italic section, and to indicate the end.

To turn on and off the underline text attribute, insert <u> in the text to indicate the beginning of an underline section, and </u> to indicate the end.

To turn on and off the ~~overstrike~~ text attribute, insert <strike> in the text to indicate the beginning of an overstrike section, and </strike> to indicate the end.

A variety of pre-defined animations and sounds can be inserted into an EMS message. These animations and sounds are defined as part of the EMS standard, and will vary slightly between different mobile phones.

Animations defined in the EMS standard include: Flirty, Glad, Skeptical, Sad, Wow, Crying, Winking, Laughing, Indifferent, In Love, Confused, Tongue Out, Angry, Glasses, and Devilish.

To insert an animation into an EMS text message, insert <animation val=xxxx/> to indicate the placement of the animation. xxxx is replaced with the name of the animation from the above list. Spaces are removed if present, for example <animation val=tongueout/> would indicate a placeholder for the "tongue out" animation.

Sounds defined in the EMS standard include: Chimes high, Chimes low, Ding, Ta Da, Notify, Drum, Claps, Fan Fare, Chords high, and Chords low.

To insert a sound into an EMS text message, insert <sound val=xxxx/> to indicate the placement of the animation. xxxx is replaced with the name of the sound from the above list. Spaces are removed if present, for example <sound val=tada/> would indicate a placeholder for the "ta da" sound.

While not widely supported by current handsets, EMS text messages can also contain colour attributes for the message text. Colours supported in the EMS standard include: black, green, red, blue, yellow, purple (magenta), cyan, gray, and white.

To apply a colour attribute to a block of text, insert <color val=xxxx> to indicate the beginning of the block of coloured text, and </color> to mark the end of a block of coloured text. xxxx can be any of the colours listed above, or it can be a numeric value between 0 and 15 to indicate a colour code as defined in the EMS specification.

EMS text messages can also include text that is larger or smaller than normal SMS text. To indicate a block of small text, insert <small> to indicate the beginning of a section of small text, and </small> to mark the end of the section. To indicate a block of large text, insert <big> to indicate the beginning of a section of large text, and </big> to mark the end of the section. Normal text does not require an indicator.

As an example, switching from large to small text would insert </large><small>, with </large> ending the large section of text, and <small> beginning the small section of text. Switching from large to normal text would insert </large>, with large ending the large section of text, implying that the text size returns to normal.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending EMS Messages – EMS Ring Tone

The "**EMS ring tone**" facility allows you to submit ring tone data in either RTTTL, iMelody or MIDI format, and send the ring tone out in either EMS, Nokia Smart Messaging, or WAP Push/MIDI format. These formats are largely supported for the sake of interfacing with older or less capable handsets, as newer handsets often support true tone formats based upon MP3.

To send a ring tone, you need to supply ring tone data. Ring tone data can be submitted either as text input, as a file via HTTP upload, or via an http:// URL reference to a file that resides on a separate web server. The ring tone data must be in RTTTL, iMelody or MIDI format.

Now Mobile does not provide technical support on the creation or deployment of ring tone services. The limited conversion options provided in the Now SMS web interface are intended as a convenience. While NowSMS may be used for the delivery of ring tone content, we strongly recommend that you evaluate other software packages to aid in the creation and conversion of ring tones.

Now Mobile also does not provide technical support or guidance regarding which ring tone formats are supported by which mobile phone models. Ring tone delivery can be a complex business, and the NowSMS product is focused on message delivery.

NowSMS supports the following input ring tone formats:

1.) **RTTTL** is the ring tone format that is used in the Nokia Smart Messaging standard. Here is a simple RTTTL example featuring the opening theme of Beethoven's Fifth Symphony:

```
fifth:d=4,o=5,b=63:8P,8G5,8G5,8G5,2D#5
```

2.) **iMelody** is the ring tone format that is used in the EMS standard. Here is a simple iMelody example featuring the opening them of Beethoven's Fifth Symphony:

```
BEGIN:IMELODY
NAME:fifth
BEAT:63
STYLE:S0
MELODY:r3*3g3*3g3*3g3*3#d1
END:IMELODY
```

3.) **MIDI** is a slightly more capable ring tone format which uses a binary file format instead of a text format. While it is possible to convert a small MIDI file into a text string of hex characters, more commonly, a MIDI file would either be submitting via HTTP file upload, or referenced via URL from another web server.

4.) It is also possible to use this facility to send ring tones of other formats out via a WAP Multimedia Message *(see page 98)*. When a ring tone file in a format other than RTTTL, iMelody or MIDI is sent via this facility, it cannot be submitted as text input, and needs to be submitted via HTTP file upload, or via an http:// URL reference to a file that resides on a separate web server. In this case, the output ring tone format must be "WAP Push", and

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

NowSMS will send the ring tone out via WAP Multimedia Push without performing any conversion of the ring tone data.

Any of the following HTTP variables can be used for specifying the input ring tone data:

RingToneDataText	EMS Ringtone	A text string that contains ring tone data in either RTTTL, iMelody or MIDI format. *(MIDI is a binary format, therefore MIDI data must be represented as a hex string when using this parameter.)*
RingToneDataFile	EMS Ringtone	Ring tone data submitting using HTTP file upload. The file can contain ring tone data in either RTTTL, iMelody or MIDI format. *(Note: This parameter can only be used in an HTTP POST with the content type of multipart/form-data.)*
RingToneDataURL	EMS Ringtone	HTTP URL pointer to a ring tone file residing on another web server. The ring tone file can be in either RTTTL, iMelody or MIDI format.

NowSMS supports the following ring tone output formats:

1.) **Nokia Smart Messaging** – This is the binary encoding for RTTTL, which was originally defined by Nokia. (Note that NowSMS currently only supports the sending these messages out in binary format. The text "//SCKL" format may be supported in a future release.)

2.) **EMS (iMelody)** – The ring tone is converted to iMelody, if necessary, and encoded as an EMS message.

3.) **EMS Short Format (iMelody without headers)** – The ring tone is converted to iMelody, if necessary. As EMS can be rather verbose, the headers are stripped from the iMelody data, and it is then encoded as an EMS message.

EMS iMelody messages are typically larger than Nokia Smart Messaging encodings because of the verbose iMelody headers. It is therefore more likely that a longer melody will be forced to span multiple SMS messages. Many EMS compatible phones do not support melodies that span multiple SMS messages, requiring the use of the EMS Short Format to attempt to fit the ring tone into a single message.

4.) **WAP Push (MIDI or no conversion)** – If the input ring tone is in RTTTL or iMelody format, it is converted to MIDI. Otherwise, no conversion is performed. The output ring tone is delivered as a WAP Multimedia Message *(see page 98)*.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The following HTTP variable settings are used for specifying the ring tone output format:

RingToneOut	EMS Ringtone	Specifies the output format to be used for the ring tone: "Nokia" (Nokia Smart Messaging) - This is the binary encoding for RTTTL, which was originally defined by Nokia. (Note that NowSMS currently only supports the sending these messages out in binary format. The text "//SCKL" format may be supported in a future release.) "EMS" (EMS - iMelody) – The ring tone is converted to iMelody, if necessary, and encoded as an EMS message. "EMSShort" (EMS Short Format - iMelody without headers) – The ring tone is converted to iMelody, if necessary. As EMS can be rather verbose, the headers are stripped from the iMelody data, and it is then encoded as an EMS message. "WAPPush" (WAP Push - MIDI or no conversion) – If the input ring tone is in RTTTL or iMelody format, it is converted to MIDI. Otherwise, no conversion is performed. The output ring tone is delivered as a WAP Multimedia Message.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending EMS Messages – EMS Picture Message

The "**EMS Picture Message**" facility allows you to send simple picture messages using either the EMS or Nokia Smart Messaging format. These formats refer to the old monochrome/black & white images that are sent via SMS, as opposed to the more advanced functionality offered by MMS. While there is little use for monochrome images in a Technicolor world, these message types can be useful for applications that need to send bar code types of images via SMS, where they cannot assume that recipient devices will have MMS or WAP capabilities.

To send a picture message , you need to supply picture or image data. Image data can be submitted either as text input, via HTTP file upload, or via an http:// URL reference to a file that resides on a separate web server. The image data must be in BMP, GIF or JPEG format. (To input a BMP, GIF or JPEG image as a text string, it must be converted to a text string of hex characters where each binary byte of the image is represented as two hex characters. File upload or referencing a web server URL that contains the image is usually easier.) Input images should have a width in pixels that is a multiple of 8.

NowSMS supports the following picture message output formats from this interface:

1.) Nokia Smart Messaging
2.) EMS
3.) WAP Multimedia Message *(see page 98)*

Keep in mind that images sent via this interface that are to be converted to Nokia Smart Messaging or EMS message should be kept small in size. For larger images, use MMS.

Any of the following HTTP variables can be used for specifying the input image data:

PictureMessageDataText	EMS Picture Message	A text string that contains image data in either BMP, JPEG or GIF format. *(These are all binary formats, therefore the image data must be represented as a hex string when using this parameter.)*
PictureMessageDataFile	EMS Picture Message	Image data submitting using HTTP file upload. The file can contain image data in either BMP, JPEG or GIF format. *(Note: This parameter can only be used in an HTTP POST with the content type of multipart/form-data.)*
PictureMessageDataURL	EMS Picture Message	HTTP URL pointer to an image file residing on another web server. The image file can be in either BMP, JPEG or GIF format.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The following HTTP variable settings are used for specifying the ring tone output format:

PictureMessageOut	EMS Picture Message	Specifies the output format to be used for the picture message: "Nokia" (Nokia Smart Messaging) - This binary encoding format was originally defined by Nokia. (Note that NowSMS currently only supports the sending these messages out in binary format. The text "//SCKL" format may be supported in a future release.) "EMS" – The image is converted to EMS format, if necessary, and encoded as an EMS message. "WAPPush" (WAP Push - no conversion) – No conversion is performed on the image, and it is delivered as a WAP Multimedia Message.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending Binary Messages

To send a binary SMS message via a menu driven interface, please see the help section titled **Web Menu Interface** on page 84. This section describes how to send a binary message programmatically via URL parameters.

To send a binary message via SMS, please refer to the specifications for the particular binary message format that you wish to send, and use the following URL format:

```
http://127.0.0.1:8800/?
PhoneNumber=xxxxxxxx&Data=00112233445566&UDH=060504030201&pid=AA&dcs
=AA&binary=1
```

For 127.0.0.1, please substitute the IP address or host name assigned to your gateway PC. (Note: 127.0.0.1 is a local loop back address that can be utilized when you are connecting to the gateway from the same computer.)

For 8800, please substitute the port number that the gateway is configured to use.

Substitute the phone number that you wish to send the SMS message to for the "xxxxxxxx" in the "PhoneNumber" parameter. Use either the local phone number format, or the international phone number format (your network provider may or may not allow you to send to international phone numbers). If the international phone number format is used, note that you must substitute "%2B" for the "+" character, because of URL escaping restrictions. For example, to send an SMS to +447778001210, use %2B447778001210 instead.

The "Data" parameter should include a string of hexadecimal digits that form the binary data content for the message.

The "UDH" parameter should include a string of hexadecimal digits that form the binary user data header for the message. Common UDH parameter settings include "06050415811581" for Nokia ring tones, "06050415821582" for Nokia operator logos, and "06050415831583" for Nokia CLI logos.

The "pid" parameter is a hexadecimal value between 0 and FF that specifies the GSM 03.40 TP-Protocol-Identifier.

The "dcs" parameter is a hexadecimal value between 0 and FF that specifies the GSM 03.38 SMS Data Coding Scheme. F5 is a common data coding scheme for most binary message formats.

The "binary" parameter should be set to "1" to tell the gateway that this is a binary message.

An example EMS message which includes a predefined EMS animation and a predefined EMS sound is shown below:

```
http://127.0.0.1:8800/?
phone=xxxxxxxx&udh=080D0200040B020007&data=00&binary=1
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Refer to specifications such as "How to Create EMS Services" on the Ericsson developer site, and "Smart Messaging Services" on the Nokia developer site for more information on binary formats for SMS messages.

The gateway includes some HTML forms to simplify the process of creating Nokia smart messages. Those HTML message forms include JavaScript commands that build the binary message parameters for submitting smart messages. Refer to the JavaScript in the corresponding HTML forms and the Nokia "Smart Messaging Services" specification for additional information.

Please note that when a "user data header" is included, the data portion of the SMS message must be encoded in binary format. Text formats cannot be mixed with a user data header.

For a complete list of URL parameters, please refer to the section **Submitting SMS Messages - URL Parameters** on page 187.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending WAP Push Messages

To send a WAP Push message via a menu driven interface, please see the help section titled **Web Menu Interface** on page 90. This section describes how to send a WAP Push programmatically via URL parameters.

WAP Push messages are specially formatted SMS messages that display an alert message to the user, and give the user the option of connecting directly to a particular URL via the mobile phone's WAP browser.

For example, an e-mail application might send an alert that tells the user they have new e-mail, with a URL link to connect directly to a WAP e-mail application.

The WAP specifications define a format for applications to create XML-based "PAP" (Push Access Protocol) documents that can be posted to an operator's "PPG" (Push Proxy Gateway), in order to deliver a WAP push message to a mobile device.

Unfortunately, the complexity of this format, and the reluctance of operators to open their "PPG" to just anyone, has made it difficult for developers to deploy "WAP Push" in their applications.

The Now SMS/MMS Gateway makes it easy to generate and deliver "WAP Push" messages. While the gateway does not support all of the options available via the PAP-based PPG interface, it does implement "WAP Push" in an elegantly simple solution.

To send a WAP Push message, use the following URL format:

```
http://127.0.0.1:8800/?
PhoneNumber=xxxxxxxx&WAPURL=name.domain/path&Text=abc+def+ghi
```

For 127.0.0.1, please substitute the IP address or host name assigned to your gateway PC. (Note: 127.0.0.1 is a local loop back address that can be utilized when you are connecting to the gateway from the same computer.)

For 8800, please substitute the port number that the gateway is configured to use.

Substitute the phone number that you wish to send the SMS message to for the "xxxxxxxx" in the "PhoneNumber" parameter. Use either the local phone number format, or the international phone number format (your network provider may or may not allow you to send to international phone numbers). If the international phone number format is used, note that you must substitute "%2B" for the "+" character, because of URL escaping restrictions. For example, to send an SMS to +447778001210, use %2B447778001210 instead.

The alert text for the WAP Push message is contained in the "Text" parameter, and utilizes the same format as described in "Sending Text Messages".

Note that there are two types of "WAP Push" messages, "Service Indication (SI)" and "Service Load (SL)". The "SL" format can be selected by including "WAPSL=1" as a URL parameter, and does not support a "Text" parameter, while the "SI" format does. (By specification, the "SL" format was designed to tell the browser to connect to a URL without

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

user intervention. However, for security reasons, most mobile phones will always display a prompt before connecting to a URL. Therefore, the lack of a text parameter makes the "SL" format considerably less user-friendly than the "SI" format, and in practice, most users will exclusively use the "SI" format.)

The URL to be pushed to the mobile device is specified in the "WAPURL" parameter. Note that the "http://" portion of the URL is not necessary and is assumed. Also note that it may be necessary to escape some URL characters, please refer to the table in the **"Sending Text Messages"** section for common characters that must be escaped.

The following parameters are supported for sending WAP Push messages:

WAPURL	WAP Push	URL to be sent in the WAP Push message.
WAPPushInitiatorURI	WAP Push, OMA OTA	Sets the WAP Push Initiator URI. For more information, refer to the Technical Bulletin titled **WAP Push or OTA: Unknown Sender**.
WAPPushFlag	WAP Push, OMA OTA	Sets the WAP Push Flag. For more information, refer to the Technical Bulletin titled **WAP Push or OTA: Unknown Sender**.
WAPSIID	WAP Push (Service Indication)	"Service Indication ID" is a text string that defines an id string to be associated with a service indication push. If a push has an "WAPSIID" associated with it, it is possible to later send a "WAPSIAction=delete" push with the same "WAPSIID" value to delete the previous push message from the device inbox. Similarly, if a mobile device receives a push message with a "WAPSIID" that matches that of a previously received push that is still in its inbox, the new push message should replace the existing push message.
WAPSIACTION	WAP Push (Service Indication)	"Signal Action" specifies the type of alert to be associated with the push. While this is not very widely supported, the general intent is to associate a priority with the alert. Valid settings are: "signal-high" - high priority alert "signal-medium" - medium priority alert "signal-low" - low priority alert "signal-none" - do not generate a

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

		notification alert for this push "delete" - if a previously sent push exists in the device inbox, with the same WAPSIID as a specified in this push, then the push should be deleted from the device inbox.
WAPSIEXPIRES	WAP Push (Service Indication)	The "SI Expires" field specifies a date/time at which the receiving device should automatically expire the push. This is a date/time value relative to GMT, in the format "yyyy-mm-ddThh:mm:ssZ". For example, "2006-02-24T00:00:00Z".
WAPSICREATED	WAP Push (Service Indication)	The "SI Created" field specifies a creation date/time stamp to be associated with the push. If specified, this date/time stamp should take the format "yyyy-mm-ddThh:mm:ssZ", specifying a date/time value relative to GMT. For example, "2006-02-24T00:00:00Z".
WAPSL	WAP Push (Service Load)	When the "WAPURL" parameter is specified, set this parameter to any value to send the WAP Push as a "Service Load" (SL) message, instead of the default "Service Indication" (SI) message.
WAPSLAction	WAP Push (Service Load)	Specifies the type of action to be taken upon receipt of a "Service Load" push. Valid settings are: "execute-low" - The browser fetches the URL and executes it in a non-intrusive manner "execute-high" - The browser fetches the URL, executes it and displays it in a manner that may be considered intrusive "cache" - The browser fetches the URL and saves the resulting data in the browser's cache *(if a cache does not exist, the push is ignored)*

For a complete list of URL parameters, please refer to the section **Submitting SMS Messages - URL Parameters** on page 187.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending Multimedia WAP Push Messages

To send a Multimedia WAP Push message via a menu driven interface, please see the help section titled **Web Menu Interface** on page 98. This section describes how to send a Multimedia WAP Push programmatically via URL parameters.

WAP Push messages are specially formatted SMS messages that display an alert message to the user, and give the user the option of connecting directly to a particular URL via the mobile phone's WAP browser.

A Multimedia WAP Push is a NowSMS feature that is designed to provide functionality similar in concept to MMS, but with the message being handled by the WAP browser instead of the MMS client. *(This is advantageous for areas where the default operator settings do not allow content from external MMSCs.)*

The way that it works is that multimedia objects (images, sound files, program files) are submitted to the NowSMS gateway. Rather than packaging the content as an MMS message, the gateway packages the content as a WAP WML page with links to the embedded objects *(and scaled down versions of the images appearing in-line)*. NowSMS sends a WAP push message over SMS to the recipient device with a URL pointer back to this dynamically created WAP WML page.

Because the URL for the Multimedia WAP Push message is dynamically generated, you can also configure NowSMS to automatically delete the dynamically generated URL link after the recipient has had a chance to retrieve the content *(helping to prevent the URL from being shared with others)*. To enable this auto-delete feature, edit MMSC.INI, and under the [MMSC] section header, add ExpireDynamicLinks=##, where ## is the number of minutes after the link is first accessed before it should be automatically deleted.

Sending a Multimedia WAP Push message via NowSMS is very similar to sending an MMS message. Generally speaking, you should follow the same steps as described in **Submitting MMS Messages to NowSMS** on page 165, except that there special flags must be set to indicate that Multimedia WAP Push is to be used instead of MMS.

For the **Now SMS/MMS Proprietary URL Submission** approach *(page 178)*, define the "MMSWAPPush=Yes" variable in the submission. This can either be done by including "&MMSWAPPush=Yes" in the URL, or you can insert another section to the multipart, which would defines the MMSWAPPush variable. For example, insert this after the MMSSubject section of the multipart in the above referenced section:

```
----boundary-border--
Content-Disposition: form-data; name="MMSWAPPush"

Yes
```

For the **MM7** approach *(page 180)*, add **/TYPE=WAPP** after the phone number of a recipient. This will tell NowSMS to send the message out as a multimedia WAP push instead of an MMS. For example, `<Number>+44777777777/TYPE=WAPP</Number>`.

For the **MM4** approach *(page 183)*, change the address in the RCPT TO: to be `<phonenumber.wappush@mms.domain.name>` or `<phonenumber/TYPE=WAPP@mms.domain.name>`.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

For **MM1** *(page 185)*, or **EAIF** *(page 186)*, specify the recipient address in the format phonenumber/TYPE=WAPP, instead of phonenumber/TYPE=PLMN which is what would be used for sending an MMS.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending WAP OTA Messages

To send a WAP OTA message via a menu driven interface, please see the help section titled **Web Menu Interface** on page 101. This section describes how to send a WAP OTA message programmatically via URL parameters.

WAP OTA (Over The Air) Messages are special SMS messages that contain information used to configure the settings of a WAP browser in a mobile phone. There are two basic types of OTA messages, the most common type of OTA message contains a complete set of configuration parameters for the WAP browser, and a second type of OTA message contains a single bookmark.

WAP Bookmark OTA Messages

The WAP Bookmark OTA message is only supported by few mobile phones (at the time this document was written). To send a WAP Bookmark OTA message, use the following URL format:

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&WAPBookmark=name.domain/path
```

For 127.0.0.1, please substitute the IP address or host name assigned to your gateway PC. (Note: 127.0.0.1 is a local loop back address that can be utilized when you are connecting to the gateway from the same computer.)

For 8800, please substitute the port number that the gateway is configured to use.

Substitute the phone number that you wish to send the SMS message to for the "xxxxxxxx" in the "PhoneNumber" parameter. Use either the local phone number format, or the international phone number format (your network provider may or may not allow you to send to international phone numbers). If the international phone number format is used, note that you must substitute "%2B" for the "+" character, because of URL escaping restrictions. For example, to send an SMS to +447778001210, use %2B447778001210 instead.

The URL to be pushed to the mobile device as a bookmark is specified in the "WAPBookmark" parameter. Note that the "http://" portion of the URL is not necessary and is assumed. Also note that it may be necessary to escape some URL characters, please refer to the table in the **Sending Text Messages** *(page 201)* section for common characters that must be escaped.

WAP Configuration OTA Messages

The gateway supports OTA documents containing "Browser settings" or "Browser bookmarks", compatible with the Nokia/Ericsson ***"Over The Air Settings Specification"***, with support up to v7.0.

This specification can be downloaded from the developer area of either the Nokia or SonyEricsson developer web sites.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Three approaches are provided for sending WAP Configuration OTA messages:

1.) The "web menu" interface of the gateway provides a menu based interface for specifying WAP configuration settings.

2.) URL parameters can be passed to the gateway to dynamically define WAP configuration settings.

3.) Manually create an OTA document based on the Nokia/Ericsson specification, and store this document in the "OTA" subdirectory of the gateway installation, or POST the document to the gateway via the web interface.

The "web menu" interface is defined in the help section titled **Web Menu Interface** on page 101. The other approaches are defined below.

WAP Configuration OTA - URL Parameters

Using the WAP Configuration OTA URL parameters requires a good understanding of the Nokia/Ericsson OTA Specification. This document specifies the mapping of gateway URL parameters to OTA setting parameters. The value provided for the gateway URL parameter is applied to the corresponding OTA setting parameter. Please refer to the Nokia/Ericsson specification for documentation of the OTA setting parameters.

```
OTA_BEARER - maps to TYPE=ADDRESS, PARM NAME=BEARER
OTA_PPP_AUTHTYPE - maps to TYPE=ADDRESS, PARM NAME=PPP_AUTHTYPE
OTA_PPP_AUTHNAME - maps to TYPE=ADDRESS, PARM NAME=PPP_AUTHNAME
OTA_PPP_AUTHSECRET - maps to TYPE=ADDRESS, PARM NAME=PPP_AUTHSECRET
OTA_PPP_LOGINTYPE - maps to TYPE=ADDRESS, PARM NAME=PPP_LOGINTYPE
OTA_PROXY - maps to TYPE=ADDRESS, PARM NAME=PROXY
OTA_PROXY_TYPE - maps to TYPE=ADDRESS, PARM NAME=PROXY_TYPE
OTA_PROXY_AUTHNAME - maps to TYPE=ADDRESS, PARM NAME=PROXY_AUTHNAME
OTA_PROXY_AUTHSECRET - maps to TYPE=ADDRESS, PARM NAME=PROXY_AUTHSECRET
OTA_PROXY_LOGINTYPE - maps to TYPE=ADDRESS, PARM NAME=PROXY_LOGINTYPE
OTA_PORT - maps to TYPE=ADDRESS, PARM NAME=PORT
OTA_CSD_DIALSTRING - maps to TYPE=ADDRESS, PARM NAME=CSD_DIALSTRING
OTA_CSD_CALLTYPE - maps to TYPE=ADDRESS, PARM NAME=CSD_CALLTYPE
OTA_CSD_CALLSPEED - maps to TYPE=ADDRESS, PARM NAME=CSD_CALLSPEED
OTA_ISP_NAME - maps to TYPE=ADDRESS, PARM NAME=ISP_NAME
OTA_SMS_SMSC_ADDRESS - maps to TYPE=ADDRESS, PARM NAME=SMS_SMSC_ADDRESS
OTA_USSD_SERVICE_TYPE - maps to TYPE=ADDRESS, PARM
NAME=USSD_SERVICE_TYPE
OTA_GPRS_ACCESSPOINTNAME - maps to TYPE=ADDRESS, PARM
NAME=GPRS_ACCESSPOINTNAME
OTA_URL - maps to TYPE=URL
OTA_MMSURL - maps to TYPE=MMSURL
OTA_NAME - maps to TYPE=NAME, PARM NAME=NAME
OTA_BOOKMARK_NAME - maps to TYPE=BOOKMARK, PARM NAME=NAME
OTA_BOOKMARK_URL - maps to TYPE=BOOKMARK, PARM NAME=URL
OTA_ID - maps to TYPE=ID, PARM NAME=NAME
```

Note that the "Send WAP OTA Settings" implementation in the gateway "web menu" interface uses this URL interface to submit OTA setting parameters. Viewing the source

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

HTML for the corresponding "web menu" interface pages may provide an improved understanding of this URL interface.

WAP Configuration OTA - OTA Documents

It is also possible to provide OTA configurations by creating one or more OTA documents that contain settings compatible with the Nokia/Ericsson specification.

OTA documents should be created in the OTA subdirectory of the gateway installation, and given a file extension of ".OTA".

Once an OTA document has been created, to send an OTA "Browser settings" file to a mobile phone, use the following URL format:

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&OTA=filename
```

The "OTA" parameter specifies the name of a file located in the OTA subdirectory of the gateway with a file extension of ".OTA". For example, in the above sample URL, the gateway would attempt to locate a file named "filename.OTA" in the OTA gateway subdirectory.

To send an OTA "Browser bookmarks" file to a mobile phone, use the following URL format:

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&OTABookmark=filename
```

The "OTABookmark" parameter uses the same format as the "OTA" parameter when sending "Browser settings", except that it expects the browser bookmark settings file to have a file extension of ".BM".

An example OTA "Browser settings" file is shown below, for additional information, please refer to the Nokia/Ericsson "Over The Air Settings Specification".

GSM/CSD Settings Example:

```
<?xml version="1.0" encoding="UTF-8"?>
<CHARACTERISTIC-LIST>
   <CHARACTERISTIC TYPE="ADDRESS">
      <PARM NAME="BEARER" VALUE="GSM/CSD"/>
      <PARM NAME="PROXY" VALUE="12.34.56.78"/>
      <PARM NAME="CSD_DIALSTRING" VALUE="+12135551212"/>
      <PARM NAME="PPP_AUTHTYPE" VALUE="PAP"/>
   </CHARACTERISTIC>
   <CHARACTERISTIC TYPE="URL"
VALUE="http://mobileinternet.ericsson.com"/>
   <CHARACTERISTIC TYPE="NAME">
      <PARM NAME="NAME" VALUE="Mobile Internet"/>
   </CHARACTERISTIC>
   <CHARACTERISTIC TYPE="BOOKMARK">
      <PARM NAME="NAME" VALUE="Mobile Internet"/>
      <PARM NAME="URL" VALUE="http://mobileinternet.ericsson.com"/>
   </CHARACTERISTIC>
</CHARACTERISTIC-LIST>
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

It is also possible to send OTA messages without creating an OTA document on the gateway by submitting an HTTP POST request to the gateway with the content of the POST message being an OTA "Browser Settings" file. It is not possible to submit such a request via a standard web browser, instead this request must be submitted programmatically. Submit the POST to a URL of:

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&OTA=POST
```

When submitting an OTA request in this manner, the HTTP POST request **must** include a "Content-Length:" header.

To provide a more complete example, to perform this HTTP POST, an application can open an HTTP connection to the port for the NowSMS web interface, and send the following data:

```
POST /?PhoneNumber=xxxxxxxx&user=username&password=password&OTA=POST HTTP/1.0
Content-Length: yyyyyy
(*blank line*)
<xml settings document>
```

When sending an XML document in this manner, the HTTP POST request must include a "Content-Length:" header. This should match the length of your XML document. This is how the server detects that the HTTP POST is complete. The "&user=" and "&password=" parameters specify an "SMS Users" account that is allowed to submit messages via NowSMS.

NowSMS will scan the XML content to automatically determine what the content type is, so that it can be encoded properly for sending over the air.

For a complete list of URL parameters, please refer to the section **Submitting SMS Messages - URL Parameters** on page 187.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending OMA Provisioning Content OTA Messages

To send an OMA Provisioning Content OTA message via a menu driven interface, please see the help section titled **Web Menu Interface** on page 112. This section describes how to send a OMA Provisioning Content message programmatically via URL parameters.

OMA Provisioning Content Messages are special SMS messages that contain information used to configure certain settings of a mobile phone, such as settings for the browser, MMS client or SyncML client.

The gateway supports OMA Provisioning Content documents compatible with the Open Mobile Alliance ***"Provisioning Content Specification v1.1***.

This specification can be downloaded from the Open Mobile Alliance web site at http://www.openmobilealliance.org.

Two approaches are provided for sending OMA Provisioning Content messages:

1.) The "web menu" interface of the gateway provides a menu based interface for specifying simple browser and MMS client configuration settings.

2.) Manually create an OTA document based on the OMA Provisioning Content specification, and store this document in the "OTA" subdirectory of the gateway installation, or POST the document to the gateway via the web interface.

The "web menu" interface is defined in the help section titled **Web Menu Interface** on page 112. The other approach is defined below.

OMA Provisioning Content documents should be created in the OTA subdirectory of the gateway installation, and given a file extension of ".OTA".

Once a document has been created, to send the document to a mobile phone, use the following URL format:

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&OMAOTA=filename
```

The "OMAOTA" parameter specifies the name of a file located in the OTA subdirectory of the gateway with a file extension of ".OTA". For example, in the above sample URL, the gateway would attempt to locate a file named "filename.OTA" in the OTA gateway subdirectory.

An example OMA Provisioning Content document for configuring browser settings on a mobile phone is shown below, for additional information, please refer to the OMA Provisioning Content Specification.

```
<wap-provisioningdoc>
        <characteristic type="BOOTSTRAP">
                <parm name="NAME" value="MoviStar Spain"/>
        </characteristic>
        <characteristic type="NAPDEF">
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
            <parm name="NAME" value="MoviStar Spain"/>
            <parm name="NAPID" value="MoviStar_Spain_NAPID"/>
            <parm name="BEARER" value="GSM-GPRS"/>
            <parm name="NAP-ADDRESS" value="wap.movistar.es"/>
            <parm name="NAP-ADDRTYPE" value="APN"/>
            <characteristic type="NAPAUTHINFO">
                  <parm name="AUTHTYPE" value="PAP"/>
                  <parm name="AUTHNAME" value="WAPTM"/>
                  <parm name="AUTHSECRET" value="WAPTM"/>
            </characteristic>
      </characteristic>
      <characteristic type="PXLOGICAL">
            <parm name="NAME" value="MoviStar Spain"/>
            <parm name="PROXY-ID" value="MoviStar Spain_Proxy"/>
            <parm name="STARTPAGE" value="http://wap.movistar.com"/>
            <characteristic type="PXPHYSICAL">
                  <parm name="PHYSICAL-PROXY-ID" value="MoviStar
            Spain_PhProxy"/>
                  <parm name="PXADDR" value="192.168.80.21"/>
                  <parm name="PXADDRTYPE" value="IPV4"/>
                  <parm name="TO-NAPID" value="MoviStar_Spain_NAPID"/>
                  <characteristic type="PORT">
                        <parm name="PORTNBR" value="9201"/>
                        <parm name="SERVICE" value="CO-WSP"/>
                  </characteristic>
            </characteristic>
      </characteristic>
      <characteristic type="APPLICATION">
            <parm name="APPID" value="w2"/>
            <parm name="TO-PROXY" value="MoviStar Spain_Proxy"/>
            <parm name="NAME" value="Browser"/>
            <characteristic type="RESOURCE">
                  <parm name="URI" value="http://wap.movistar.com"/>
                  <parm name="STARTPAGE"/>
            </characteristic>
      </characteristic>
</wap-provisioningdoc>
```

It is also possible to send Provisioning Content messages without creating a document on the gateway by submitting an HTTP POST request to the gateway with the content of the POST message being the Provisioning Content document. This document can either be sent programmatically, or it can be sent via the "Send XML Settings" option in the web menu interface. To submit the document via HTTP POST, it should be submitted to a URL of:

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&OTA=POST
```

When submitting an OTA request in this manner, the HTTP POST request **must** include a "Content-Length:" header.

To provide a more complete example, to perform this HTTP POST, an application can open an HTTP connection to the port for the NowSMS web interface, and send the following data:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
POST /?PhoneNumber=xxxxxxxx&user=username&password=password&OTA=POST HTTP/1.0
Content-Length: yyyyyy
(*blank line*)
<xml settings document>
```

When sending an XML document in this manner, the HTTP POST request must include a "Content-Length:" header. This should match the length of your XML document. This is how the server detects that the HTTP POST is complete. The "&user=" and "&password=" parameters specify an "SMS Users" account that is allowed to submit messages via NowSMS.

NowSMS will scan the XML content to automatically determine what the content type is, so that it can be encoded properly for sending over the air.

The URL request can also include an OTAPIN parameter specifying the PIN associated with the request, and an OTAPINTYPE parameter specifying the type of PIN associated with the request (USERPIN or NETWPIN).

For a complete list of URL parameters, please refer to the section **Submitting SMS Messages - URL Parameters** on page 187.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending XML Settings Documents and Objects

To send an XML Settings Document via a menu driven interface, please see the help section titled **Web Menu Interface** on page 118. This section describes how to send a OMA Provisioning Content message programmatically via HTTP POST.

The "Send XML Settings Document" interface supports the following types of XML settings documents:

Nokia/Ericsson Over The Air Settings (OTA) Specification up to and including v7.1

Browser and MMS settings use the root XML element <CHARACTERISTIC-LIST>
SyncML settings use the root XML element <SyncSettings>
Wireless Village/IMPS settings use the root XML element <WVSettings>

Examples of this format can be generated using the **Send WAP OTA Settings** web form, which is described in more detail beginning on page 101.
Copies of the specification that documents these settings can be downloaded from either the Nokia or SonyEricsson developer web sites.

OMA (Open Mobile Alliance) Provisioning Content

All settings use the root XML element <wap-provisioningdoc>

Examples of this format can be generated using the **Send OMA Settings** web form, which is described in more detail beginning on page 112.

The OMA Provisioning Content Specification is available for download from the Open Mobile Alliance web site at http://www.openmobilealliance.org.

OMA (Open Mobile Alliance) DRM Rights Objects

The objects use the root XML element <o-ex:rights>.

The DRMCOMP utility can be used to generate DRM rights objects. For additional information, refer to **Digital Rights Management**, beginning on page 320.

The OMA DRM Right Expression Language specification can be downloaded from the Open Mobile Alliance web site at http://www.openmobilealliance.org.

WAP Push Service Indication, Service Load and Cache Operation

The operations use the root XML element <si>, <sl> or <co>.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

These formats are defined in the WAP Service Indication (WAP-167), Service Load (WAP-168), and Cache Operation (WAP-175) specifications respectively.

These specifications can all be downloaded from the Open Mobile Alliance web site at http://www.openmobilealliance.org.

OMA (Open Mobile Alliance) E-Mail Notification (EMN)

This format uses the root XML element <emn>.

The OMA E-Mail Notification specification can be downloaded from the Open Mobile Alliance web site at http://www.openmobilealliance.org.

HTTP POST Format

It is possible to send any of the supported XML Settings Documents or Objects by submitting an HTTP POST request to the NowSMS gateway with the content of the POST message being the XML document. This document can either be sent programmatically, or it can be sent via the "Send XML Settings" option in the web menu interface. To submit the document via HTTP POST, it should be submitted to a URL of:

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&OTA=POST
```

When submitting an OTA request in this manner, the HTTP POST request **must** include a "Content-Length:" header.

To provide a more complete example, to perform this HTTP POST, an application can open an HTTP connection to the port for the NowSMS web interface, and send the following data:

```
POST /?PhoneNumber=xxxxxxxx&user=username&password=password&OTA=POST HTTP/1.0
Content-Length: yyyyyy
(*blank line*)
<xml settings document>
```

When sending an XML document in this manner, the HTTP POST request must include a "Content-Length:" header. This should match the length of your XML document. This is how the server detects that the HTTP POST is complete. The "&user=" and "&password=" parameters specify an "SMS Users" account that is allowed to submit messages via NowSMS.

NowSMS will scan the XML content to automatically determine what the content type is, so that it can be encoded properly for sending over the air.

The URL request can also include an OTAPIN parameter specifying the PIN associated with the request, and an OTAPINTYPE parameter specifying the type of PIN associated with the request (USERPIN or NETWPIN).

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

OTA PIN and OTA PIN Type

An "**OTA PIN**" can be associated with an OMA Provisioning Content message, in addition to some other settings message types, to provide a layer of authentication to the message. Many devices will allow you to send OTA settings without a PIN, but some will require a PIN to be present before the settings will be accepted.

There are three different types of OTA PINs, depending on the "**OTA PIN Type**" setting.

1.) The simplest "OTA PIN Type" is "User PIN" (USERPIN). This setting indicates that a short PIN code (often 4 digits) is supplied as the "OTA PIN". When the user receives the OTA settings message, they will need to supply this PIN code in order to be able to open the message and apply the settings.
2.) "Network PIN" (NETWPIN) indicates the PIN is a network PIN code. In the GSM environment, this is the IMSI number associated with the SIM card in the device. (Hint, if you want to experiment with determining the PIN card associated with a SIM, you can put the SIM into a GSM modem and the AT+CIMI command to return the IMSI. However, not all GSM modems support the AT+CIMI command.) When the device receives the settings, if the NETWPIN does not match the IMSI, the settings will be discarded.
3.) An additional type of PIN, known as "USERNETWPIN" also exists, which indicates a combination of the USERPIN and NETWPIN types. To use this OTA PIN type, select "Network PIN", and define the OTA PIN as the IMSI number associated with the SIM card in the device, followed by a ":" character, followed by a USERPIN *(e.g., 1234567889012345:1234)*. When the device receives the settings, the user will be prompted for a PIN. This user supplied PIN, and the SIM card IMSI, must match in order for the settings to be accepted.

The URL request can include an OTAPIN parameter specifying the PIN associated with the request, and an OTAPINTYPE parameter specifying the type of PIN associated with the request (USERPIN, NETWPIN or USERNETWPIN). For example:

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&OTA=POST&OTAPIN=1234&OTAPINTYPE=USERPIN
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending MMS Notifications and Content

To send an MMS message via a menu driven interface, please see the help section titled **Web Menu Interface** on page 97. This section describes how to send an MMS message programmatically via URL parameters. Specifically this section details the process of sending an MMS notification.

An MMS Notification is used when you have pre-packaged MMS message content residing on an existing web server, and you want to simply use NowSMS to send an MMS notification to tell an MMS compatible client to retrieve the content.

Please note that this technique will not work in many mobile operator environments because of firewalls deployed in the mobile operator network. However, it is described here for developers who want to better understand the MMS delivery process.

As an alternative, it is possible to submit your complete MMS message content to the NowSMS server. This technique is described in **Submitting MMS Messages to NowSMS** on page 165.

MMS (Multimedia Messaging Service) messages are sent using a combination of SMS and WAP technologies. When an MMS message is sent, a mobile device receives an MMS notification message via SMS. When this MMS notification message is received by the mobile device, the mobile device automatically initiates a WAP gateway connection to download the content of the MMS message.

To send an MMS message, you must first create an MMS message file. The format of an MMS message file is documented in the MMS Encapsulation Protocol specification published by the Open Mobile Alliance (http://www.openmobilealliance.org) and/or the WAP Forum (http://www.wapforum.org). The MMS message file format consists of an MMS message binary header, followed by a multipart MIME message where the multipart message is encoded in a binary multipart format as defined by the WAP Wireless Session Protocol (WSP) specification. This binary MMS message file is stored on a web server using a MIME type of application/vnd.wap.mms-message and an MMS message type of m-retrieve-conf. A subset of the binary MMS header is sent as an MMS notification message (MMS message type m-notification-ind) via SMS to the mobile device together with a URL pointer to the location of the complete message.

The gateway includes an MMS message compiler to assist in the creation of the MMS message files, which will be described shortly. It is also possible to create MMS message files by uploading the individual MMS message components via the gateway web menu interface (page 94) or by submitting the complete MMS message content to the NowSMS server via a variety of different protocols as described in **Submitting MMS Messages to NowSMS** on page 165. This section of the document focuses more on the programmatic creation of MMS message files.

Once an MMS message file has been built and published via a web server, the MMS notification message can be sent by the gateway using the following URL format:

```
http://127.0.0.1:8800/?
PhoneNumber=xxxxxxxx&MMSURL=name.domain/path/filename.mms
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

For 127.0.0.1, please substitute the IP address or host name assigned to your gateway PC. (Note: 127.0.0.1 is a local loop back address that can be utilized when you are connecting to the gateway from the same computer.)

For 8800, please substitute the port number that the gateway is configured to use.

Substitute the phone number that you wish to send the SMS message to for the "xxxxxxxx" in the "PhoneNumber" parameter. Use either the local phone number format, or the international phone number format (your network provider may or may not allow you to send to international phone numbers). If the international phone number format is used, note that you must substitute "%2B" for the "+" character, because of URL escaping restrictions. For example, to send an SMS to +447778001210, use %2B447778001210 instead.

The URL that contains the MMS message file is specified in the "MMSURL" parameter. Note that the "http://" portion of the URL is not necessary and is assumed. Also note that it may be necessary to escape some URL characters, please refer to the table in the **Sending Text Messages** section *(page 201)* for common characters that must be escaped. Before sending the MMS notification message, the gateway will validate that the MMS message file is of the MIME type application/vnd.wap.mms-message, and is of the MMS message type m-retrieve-conf.

Additional parameters supported for the MMS notification message include "MMSFROM" and "MMSSUBJECT", which can be used to override the message sender and subject in the MMS message file.

For a complete list of URL parameters, please refer to the section **Submitting SMS Messages - URL Parameters** on page 187.

Creating MMS Message Files – MMSCOMP

As the MMS message file format is a binary file format, special tools are required to create MMS message files. The MMSCOMP utility is provided to assist in the creation of MMS message files. The MMSCOMP utility accepts text input files to create a binary MMS Message file.

A standard format is not defined for a text version of an MMS message file, however a format can easily be derived based upon the MMS Encapsulation Protocol Specification. The MMSCOMP utility accepts as input a file that contains text representations of the MMS header, and one or more files (image, sound, text, etc.) to comprise the multipart message content.

MMSCOMP is a command-line utility that accepts the following command-line format:

MMSCOMP [-ccharset] header.file [data1.file [data2.file [data3.file …]]]

-ccharset is used to specify a character set for the text components of the input files. This parameter is not required. The default character set is iso-8859-1. Other supported character sets include big5, iso-10646-ucs-2, iso-8859-1, iso-8859-2, iso-8859-3, iso-8859-4, iso-8859-5, iso-8859-6, iso-8859-7, iso-8859-8, iso-8859-9, shift_JIS, us-ascii, and utf-8.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

header.file is a text file that contains text representations of the MMS message header. Supported MMS message headers include:

```
X-Mms-Message-Type: m-retrieve-conf   (required)
X-Mms-Transaction-Id: text-string
X-Mms-Version: 1.0
Message-Id: text-string (usually x@x format)
Date: HTTP-date-format
From: address@domain or +InternationalPhoneNumber/TYPE=PLMN (Address-present-token is
assumed)
To: address@domain or +InternationalPhoneNumber/TYPE=PLMN  (use multiple headers for
multiple recipients)
Cc: (same format as To)
Bcc: (same format as To)
Subject: text-string
X-Mms-Message-Class: Personal, Advertisement, Informational or Auto (default is Personal)
X-Mms-Priority: Low, Normal or High (default is Normal)
X-Mms-Delivery-Report: Yes or No (default is No)
X-Mms-Read-Reply: Yes or No (default is No)
Content-type: MIME-Type (default is application/vnd.wap.multipart.related, override
default with caution!)
X-NowMMS-Content-Location: filename;content-type (optional, use multiple headers for
multiple files)
```

Only the X-Mms-Message-Type header is required, other headers are optional. It is recommended that From and Subject headers always be included.

Note that while the message may contain multiple recipients in the To, Cc and Bcc headers, the gateway itself will only send the MMS notification message to one recipient at a time, as specified in the PhoneNumber parameter passed in a URL request.

At least one data file must be specified to provide the content of the MMS message. This data file can be specified on the command line (e.g., data1.file, data2.file, data3.file, …), or it may be specified in the MMS header file with one or more X-NowMMS-Content-Location headers.

If the first data file is a SMIL (Synchronized Multimedia Integration Language) file, then MMSCOMP will automatically parse all "src" references in the SMIL file and include any referenced files in the MMS multipart message file automatically.

If a SMIL file is to be included for presentation of the MMS message, it is recommended that the SMIL file always be specified as the first data file to the MMSCOMP command.

MMSCOMP determines the MIME type of each file based on the file extension, or when using the "X-NowMMS-Content-Type" header, the content type can be specified following the file name. File extensions of .jpg, .jpeg (image/jpeg), .gif (image/gif), .txt (text/plain), .wbmp (image/vnd.wap.wbmp) and .smil (application/smil) are recognized automatically. Other file extensions are read from the MMSCTYPE.INI file, or the Windows registry, under the registry key HKEY_CLASSES_ROOT\.extension, where ".extension" is the extension of the file. For best results, please ensure that any file types that you are using with MMSCOMP are defined in the MMSCTYPE.INI file.

The output of the MMSCOMP command will be stored in a file that matches the name of the input header file, but with ".MMS" as the file extension.

Example:

Assume that:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

1.) You have created an MMS message header file named "test.hdr".
2.) You have created a SMIL file named "testfile.smil". The "testfile.smil" file references three external files through the following references in the SMIL file:
```
<img src="image.jpg" region="Image"/>
<audio src="sound.amr"/>
<text src="text.txt" region="region1_1"/>
```
3.) The "image.jpg", "sound.amr" and "text.txt" files referenced by the "testfile.smil" file are located in the same directory as the "testfile.smil" file.

To create a binary MMS file, run:

```
MMSCOMP test.hdr testfile.smil
```

or

```
MMSCOMP test.hdr testfile.smil image.jpg sound.amr text.txt
```

If you want to specify a character set for the text file, include the –cCHARSET parameter:

```
MMSCOMP -cUTF-8 test.hdr testfile.smil
```

The output of the MMSCOMP file will be "test.mms" (e.g., the same filename as "test.hdr", but with a ".mms" file extension).

To send the compiled MMS file, you can either:
1.) Submit it as a content file to the Send MMS Message option in the Web Menu Interface *(page 94)*.
2.) Store the file on a web server using a MIME content type of "application/vnd.wap.mms-message", and use the gateway to send an MMS Notification Message (page 230).
3.) If the message recipient is defined to the MMSC built into the gateway, attach the file to an e-mail message and send the message to username@mmsdomainname, where "username" is the alias name defined for the user on the "MMSC Users" dialog, and "mmsdomainname" is the "Domain Name for MMS E-Mail" defined on the "MMSC" dialog. Please refer to the section titled **MMSC Messaging Server** *(page 122)* for information on configuring the MMSC to send and receive e-mail messages.

The MMSCOMP utility works well in conjunction with tools such as the SonyEricsson MMS Composer. The SonyEricsson MMS Composer creates SMIL (Synchronized Multimedia Integration Language) files, which are often used in MMS messages, but it does not create the complete binary MMS message file. To use the output of the SonyEricsson MMS Composer, use File/Export to export your MMS message to a specified directory. The composer will output a SMIL file and any included MMS message components to the directory specified. Create an MMS message header file, and then run MMSCOMP passing the name of the MMS message header file and the name of the SMIL file output by the SonyEricsson MMS Composer.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending Voice Mail Notification Messages

To send an SMS voice mail notification message via a menu driven interface, please see the help section titled **Web Menu Interface** on page 121. This section describes how to send a voice mail notification message programmatically via URL parameters.

Voice Mail Notification Messages are special SMS messages that are used to tell the user that they have voice mail waiting. On most mobile phones, the phone displays a message prompt, and the user can press a single key to be transferred to voice mail. This voice mail phone number is configurable via the mobile phone settings.

Voice Mail Notification Messages would be most often used in conjunction with voice mail systems. For example, a user may wish to combine their mobile phone voice mail with their office voice mail in a single voice mailbox. One way of accomplishing this is to configure the mobile phone to forward to an office phone number when the mobile phone is busy or unavailable, instead of the standard setting of forwarding to the mobile voice mail system (this setting can be configured via the mobile phone). If the user is unavailable, the office voice mail system assumes responsibility for accepting a voice mail message. The office voice mail system would be configured to make a request via the SMS gateway to turn on and off voice mail notifications for the mobile phone user.

```
http://127.0.0.1:8800/?PhoneNumber=xxxxxxxx&VoiceMail=On
```

For 127.0.0.1, please substitute the IP address or host name assigned to your gateway PC. (Note: 127.0.0.1 is a local loop back address that can be utilized when you are connecting to the gateway from the same computer.)

For 8800, please substitute the port number that the gateway is configured to use.

Substitute the phone number that you wish to send the SMS message to for the "xxxxxxxx" in the "PhoneNumber" parameter. Use either the local phone number format, or the international phone number format (your network provider may or may not allow you to send to international phone numbers). If the international phone number format is used, note that you must substitute "%2B" for the "+" character, because of URL escaping restrictions. For example, to send an SMS to +447778001210, use %2B447778001210 instead.

The following parameters are supported for turning on an off voice mail (and other related) notifications:

VoiceMail	Voice Mail Notification	"On" - Turn on voice mail waiting indication "Off" - Turn off voice mail waiting indication "FaxOn" - Turn on fax message waiting indicator "FaxOff" - Turn off fax message waiting indicator "EmailOn" - Turn on e-mail message waiting indicator "EmailOff" - Turn off e-mail message

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

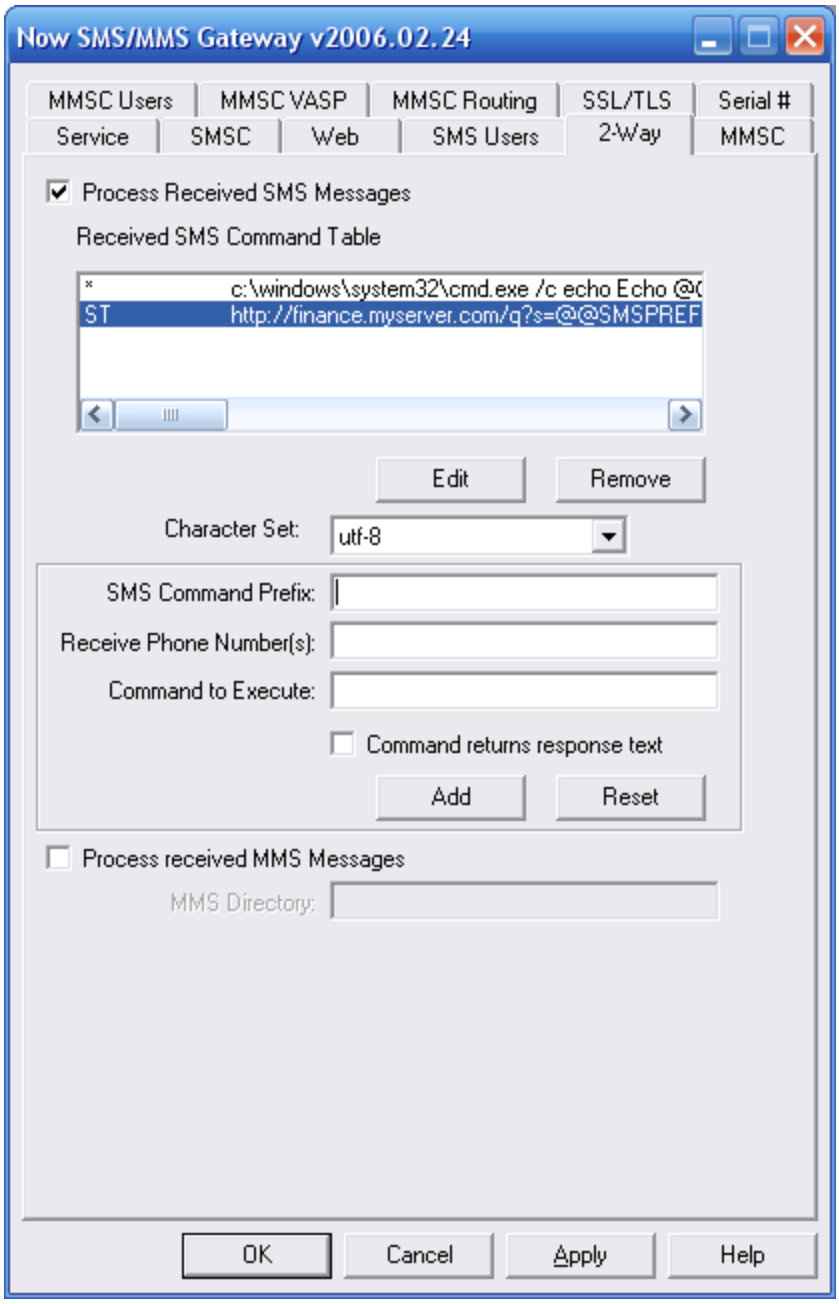
		waiting indicator "VideoOn" - Turn on video message waiting indicator "VideoOff" - Turn off video message waiting indicator "OtherOn" - Turn on other message waiting indicator "OtherOff" - Turn off other message waiting indicator
VoiceMailMessageCount	Voice Mail Notification	Specifies an optional "message count" for the number of messages waiting associated with this notification.

For a complete list of URL parameters, please refer to the section **Submitting SMS Messages - URL Parameters** on page 187.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

2-Way SMS Support

The "**2-Way**" configuration dialog contains settings relevant to the creation of 2-way applications that can receive SMS messages, and return a response based upon the content of the received SMS message.



The "**Process Received SMS Messages**" checkbox must be checked in order to enable the gateway to receive and process SMS messages.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

When an SMS message is received, the gateway will evaluate the content of the message, and can either execute a program, or connect to an HTTP URL, based upon the content of the message. The decision of how to process a received message is based upon the first "word" of the received SMS message. In the terminology of the gateway, this first word of the received message is called the "**SMS Command Prefix**". Based upon this "SMS Command Prefix", the gateway will execute a command associated with the prefix. If the received prefix does not match any defined prefix, then it is considered to be a match for a special wildcard prefix, denoted as "*".

The 2-way command processor also looks at the phone number to which the message is addressed, that is to say, the receiving (or recipient) phone number. For example, if the message was received by a GSM modem connected to the NowSMS server, the "**Receive Phone Number**" would be the phone number associated with that modem. This way different 2-way commands can be defined for processing messages that arrive via different SMSC connections.

When a command is executed based upon the receipt of an inbound message, the command line for the program or HTTP request can include replaceable parameters from the contents of the SMS message. The following replaceable parameters are supported:

@@SENDER@@	The phone number of the sender of the SMS Message.
@@SMSPREFIX@@	The first word of the SMS message.
@@SMS@@	The content of the SMS message, except the first word of the message.
@@FULLSMS@@	The complete content of the SMS message.
@@RECIP@@	If available, the phone number that is intended to receive (or did receive, if the message was received via a GSM modem connection) this message.
@@MESSAGEID@@	The local, NowSMS defined ID of the message file in the SMS-IN directory.
@@RECEIPTMESSAGEID@@	If this message is a delivery receipt, this is the message id of the originally sent message. Otherwise, this parameter is blank.
@@SERVICETYPE@@	If this message was received via SMPP, this will contain the service_type value associated with the message. Otherwise, this parameter is blank.
@@SMSCROUTE@@	Identifies the SMSC connection from which this message was received. This will either be the name of the SMSC connection (e.g., Bluetooth Modem or SMPP – a.b.c.d:xyz), or if the SMSC specific section of the SMSGW.INI has a RouteName= setting associated with the SMSC, this parameter will have the value of the RouteName= setting.
@@MSGDATE@@	Date on which the message was received by NowSMS in YYYYMMDD format.
@@MSGTIME@@	Time at which the message was received by

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

	NowSMS in HHMMSS format.
@@BINARY@@	Set to "1" if the message is in binary format, "0" otherwise. (Note: Binary messages will only be routed to a 2-way command if the "Command to Execute" is HTTP based, and @@BINARY@@ is present in the "Command to Execute" field.)
@@UDH@@	User Data Header of the received message. (Only for binary messages, see @@BINARY@@ parameter definition.)
@@PID@@	Protocol ID field (PID) of the received message. (Only for binary messages, see @@BINARY@@ parameter definition.)
@@DCS@@	Data Coding Scheme (DCS) of the received message. (Only for binary messages, see @@BINARY@@ parameter definition.)

To return results back to the user, the command can either return a simple text response directly to the gateway ("**Command returns response text**" is checked), or the command can generate a more complex response to the gateway via a separate HTTP request to the gateway. An executable program returns a simple text response to the gateway by printing results to the screen, where the gateway captures the screen output, and generates an SMS response to send the screen output text back to the sender via SMS. An HTTP request returns a simple text response to the gateway by returning content of the MIME type "text/plain".

The example dialog above illustrates a simple command that can be used for testing this 2-way capability.

Define an "**SMS Command Prefix**" of "*" (wildcard), or any prefix of your choosing.

Define "**Command to Execute**" as "c:\windows\system32\cmd.exe /c echo Echo @@FULLSMS@@".

Check "**Command returns response text**".

Press "**Add**" to add the command to the "**Received SMS Command Table**".

When an SMS is received that matches this SMS command prefix (in the case of "*", any SMS that doesn't match another defined prefix), the gateway launches a command processor (CMD) that simply echoes the text back to the screen adding the word "Echo" to the beginning of the received text. In this example, the sender of the SMS message will receive an "Echo" back of the command that they sent in to the gateway. While not an extremely useful command, this is a useful way of testing to see that the gateway is alive and capable of receiving SMS messages.

In addition to supporting the launch of command line programs, the "**Command to Execute**" field can also be an HTTP command, causing the gateway to connect via HTTP to another application server to inform the application server of details regarding the received message. For example, http://server:port/path?sender=@@SENDER@@&message=@@FULLSMS@@.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

When an HTTP command is used, if the command is to return a response to the gateway directly, the HTTP response must be of the MIME content type "text/plain".

It is also possible for any HTTP command to return an HTTP redirect response to the gateway, which instructs the gateway to fetch an alternative URL, even a URL command that contains parameters to tell the gateway to submit a message. This can be useful for creating a 2-way command script that responds with binary message content.

If an HTTP command requires HTTP authentication with a username and password, the URL format of "http://username:password@host.name/path" is supported. When a URL command is defined in this format, the gateway will connect to "http://host.name/path" using an authorization header of the specified username and password.

The **"Command to Execute"** field can also specify an e-mail address, in which case any received SMS messages that match the configured "SMS Command Prefix" will be forwarded to the specified e-mail address. To specify an e-mail address for the "Command to Execute", use the format "mailto:user@domain.com".

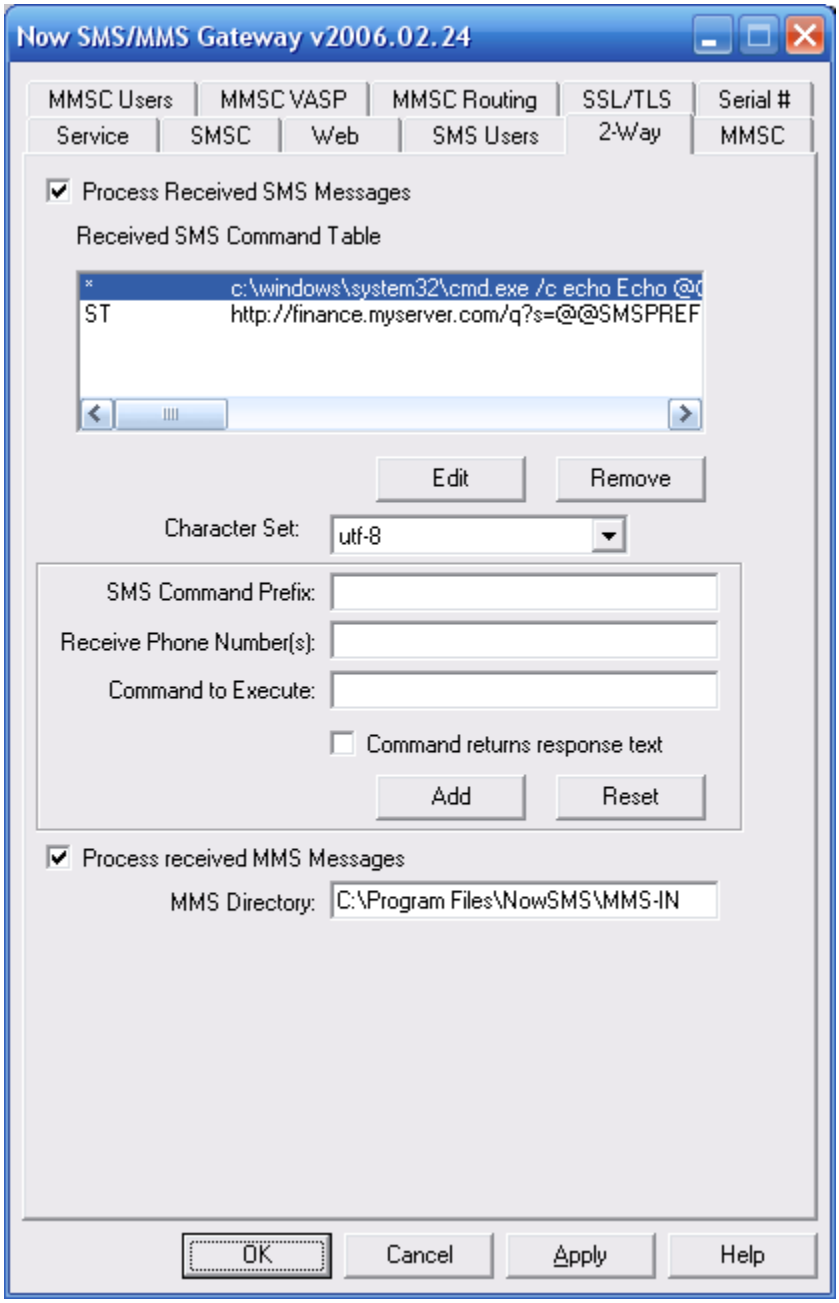
If the wildcard SMS command prefix is not associated with any command, any inbound SMS messages that do not match a prefix will be saved to the SMS-IN directory with a file extension of ".SMS", and they may be processed by another application independent of the gateway.

Some troubleshooting tips for 2-way commands, as well as simple examples using PHP, ASP and Windows scripting can be found on the NowSMS discussion board at http://www.nowsms.com/discus/messages/1/4520.html.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

2-Way MMS Support

Support for processing received MMS messages is slightly more complex than received SMS messages, because the content of MMS messages is more complex, and the network configuration information to receive MMS messages is also considerably more complex. To enable received MMS messages to be processed, on the "**2-Way**" configuration dialog, both "**Process Received SMS Messages**" and "**Process Received MMS Messages**" must be checked.



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The Now SMS/MMS Gateway can receive MMS messages via two different types of connections.

1.) MMS messages can be received via a direct connection to an operator MMSC using one of the supported protocols, including MM7, MM4 or EAIF. When any of these protocols are used, the operator MMSC will automatically connect to your Now SMS/MMS Gateway to deliver messages. For more information on this type of connection, please refer to **Connecting to an Operator MMSC – Receiving MMS Messages** on page 163.
2.) MMS messages can be received using a combination of SMS and WAP technologies. When an MMS message is sent to a mobile phone, the mobile phone first receives an MMS notification message over SMS. The Now SMS/MMS Gateway can receive this MMS notification over SMS, and then retrieve the content of the MMS message over WAP or the internet, just as it would be received by a mobile device. For more information on this type of connection, please refer to **Connecting to an Operator MMSC – Using a GPRS Modem** on page 136.

There are different options for the processing of received MMS messages. Because MMS messages typically contain multiple objects, as opposed to the simple text string of an SMS message, they are routed to applications via different interfaces.

The routing choices for received MMS messages is configured separately for each potential connection through which the gateway can receive MMS messages.

When the gateway is receiving MMS messages via an operator MMSC, an account is defined in the "MMSC VASP" configuration dialog, and the option for how to route MMS messages received over this connection is defined as part of the account definition.

When the gateway is receiving MMS messages via an SMSC or GSM modem connection, the "Properties" dialog for the SMSC configuration will include an "MMS Settings" dialog.

The choices for how received MMS messages are processed are:

"Receive to MMS-IN Directory" - In this case, the MMS message is parsed into text format, and individual components of the message are extracted into separate files. A ".hdr" file is written to the MMS-IN directory which contains a text version of the MMS header. This header file includes "X-NowMMS-Content-Location:" headers that point to the different file components that are included in the MMS message (text, images, etc.). These additional file components are stored in a subdirectory, and the location specified in the header is relative to the MMS-IN directory. The format of these ".hdr" files is consistent with the format used by the MMSCOMP utility, which is described in the section titled **Creating MMS Message Files – MMSCOMP** *(page 231)*.

"Route via MM7" - MM7 is an XML/SOAP *(see page 180)* interface that is defined by the 3GPP as a format for applications to send/receive MMS messages. When this type of routing is defined for received MMS messages, NowSMS reformats the MMS message into MM7 format, and performs an HTTP POST of the content to the specified MM7 connection. You must first define an MM7 connection on the "MMSC Routing" configuration dialog before you will be able to select this option for routing a received MMS message.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

In addition to supporting an HTTP POST in MM7 format, NowSMS supports an HTTP POST using "HTTP File Upload", where the content of the **MMS message can be processed by a PHP script**. This type of script is described in more detail on page 391.

"**Forward to E-Mail Address**" - The MMS message is converted into an e-mail message (with components of the MMS message converted to attachments) and sent to a specified e-mail address.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

E-Mail to SMS/MMS Connectivity

NowSMS has a considerable amount of flexibility for routing messages between e-mail and SMS or MMS. However, the configuration can be somewhat complex because of controls that are implemented to make sure that users do not inadvertently setup a potentially expensive open SMTP to SMS or MMS relay.

NowSMS has the ability to function as a simple SMTP server, receiving e-mail messages and routing them to SMS or MMS recipients.

NowSMS also has the ability to function as a simple POP3 server, routing received SMS or MMS messages back to an e-mail account.

It is best to configure NowSMS first so that you can send messages from the "Web Menu Interface" to confirm that NowSMS is correctly configured before you progress to the e-mail configuration.

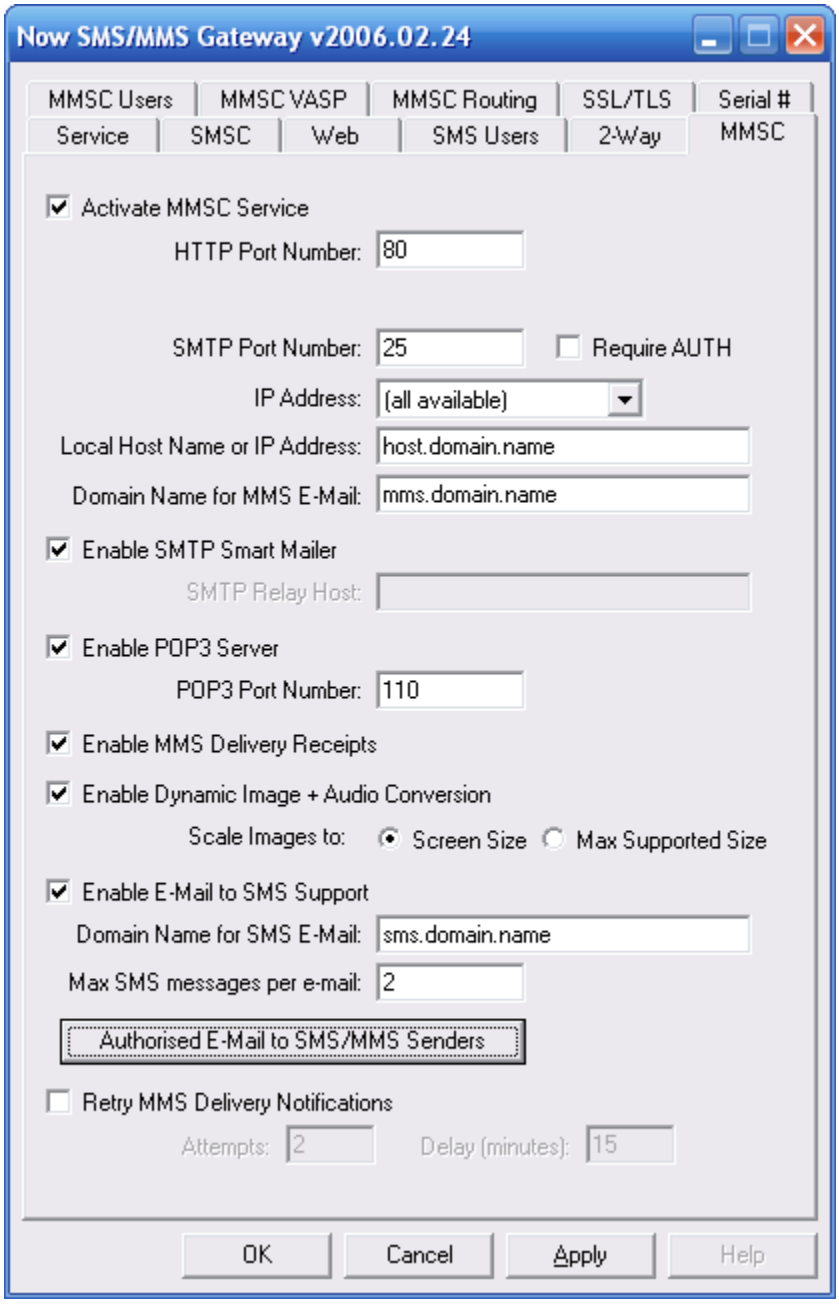
It is possible to setup NowSMS so that it only route SMTP messages to SMS or MMS from local applications on your network, or you can make it internet accessible for messages to be routed to selected SMS or MMS addresses.

The SMTP server is a component of the "MMSC" service. Even if you are not using any of the other MMS related functionality, you must activate the MMSC service in order to enable e-mail to SMS and/or MMS support.

The SMTP and e-mail related settings are configured on the "MMSC" page of the NowSMS configuration dialog. This section will describe these e-mail related settings. If you have any questions regarding "MMSC" settings that are not covered in this section, please refer to **Configuring the MMSC** beginning on page 122.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Configuring the SMTP Server



As mentioned previously, the SMTP server is a component of the "MMSC" service. Even if you are not using any of the other MMS related functionality, you must check "**Activate MMSC Service**" in order to enable e-mail to SMS and/or MMS support.

Unless you have a very specialised setup, you should configure the "**SMTP Port Number**" as "25". (Note that only one service can bind to a particular port number on a single machine. So you can have difficulty installing NowSMS on the same PC as an existing SMTP mail server. If your PC has multiple IP addresses, NowSMS does have the ability to bind to only

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

one of the available IP addresses if you select that address in the "IP Address" field beneath the "SMTP Port Number" setting. However, any other SMTP server that you run on the same PC must have similar capabilities to bind to only one IP address in order to run multiple SMTP servers on the same PC. If you are confused by all of this, the just avoid trying to run NowSMS with this SMTP support on the same PC as an another mail server.)

Skip the "**Require AUTH**" setting for now, as it will be explained later in this section.

"**Local Host Name or IP Address**" is the host name that is presented when this host communicates with another SMTP server *(HELO command)*. This host name is also used for constructing URLs when sending MMS using NowSMS as a direct delivery MMSC. Under normal conditions it should be a host name defined in the internet DNS, which resolves to the PC running NowSMS. It could also be the IP address of the PC.

"**Domain Name for MMS E-Mail**" defines the domain name that is to be used to identify messages that are to be routed as MMS messages. If the NowSMS SMTP server receives a message addressed to username@mms.domain.name where "mms.domain.name" is the domain name defined in this field, then NowSMS will attempt to route the message to the recipient via MMS. If you want to be able to send to this domain over the internet, you must define a Mail eXchange (MX) record in internet DNS that resolves this domain name to the PC that is running NowSMS.

If you are not using the MMS facilities of NowSMS, it is possible to enter a dummy value into this field.

The "Enable SMTP Smart Mailer" and "SMTP Relay Host" settings are used for routing from SMS or MMS to e-mail. These settings are not used for routing e-mail to SMS or MMS.

If "**Enable SMTP Smart Mailer**" is checked, the MMSC will perform DNS lookups to locate remote recipients and perform SMTP e-mail delivery. If "Enable SMTP Smart Mailer" is not checked, it is required that an "**SMTP Relay Host**" be defined. When the smart mailer is not enabled, the MMSC will connect to the "SMTP Relay Host" to send all outbound SMTP e-mail messages. If you are using an "SMTP Relay Host", please define an appropriate SMTP mail server in your network that will perform this SMTP message relay capability.

Checking "**Enable POP3 Server**" enables the POP3 Server. The POP3 server allows user accounts defined in the "SMS Users" dialog to connect via the POP3 e-mail protocol in order to receive SMS or MMS messages *(the accounts can also connect via the SMTP e-mail protocol to send SMS or MMS messages)*. The standard "**POP3 Port Number**" is 110, however it is possible to use a non-standard port, if desired. The POP3 server will be described in more detail later in this section.

The "Enable MMS Delivery Receipts" and "Enable Dynamic Image + Audio Conversion" options can be ignored, as they are not relevant to e-mail connectivity.

To enable e-mail to SMS support, check "**Enable E-Mail to SMS Support**".

Next, specify a "**Domain Name for SMS E-Mail**". This defines the domain name that is to be used to identify messages that are to be routed as SMS messages. If NowSMS receives a message addressed to username@sms.domain.name, where "sms.domain.name" is the domain name defined in this field, then NowSMS will attempt to route the message to the recipient via SMS. If you want to be able to send to this domain over the internet, you

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

must define a Mail eXchange (MX) record in internet DNS that resolves this domain name to the PC that is running NowSMS.

Note that the "Domain Name for SMS E-Mail" must be different from the "Domain Name for MMS E-Mail". If the values are the same, NowSMS will attempt to route any e-mail for that domain via MMS instead of SMS.

"Max SMS messages per e-mail" specifies the maximum number of SMS messages to be sent out for a single received e-mail message. If a message contains only characters in the standard GSM character set, you can fit 160 characters in a single message *(or only 70 characters when any Unicode characters outside of this character set are included in a message)*. However, NowSMS can send out a set of concatenated messages which will appear as a single longer message on most handsets. Set this value to "1" if you do not want to use any concatenated/long SMS, or set a value to specify the maximum number of SMS to be sent in a concatenated SMS for a single e-mail message. NowSMS will truncate any messages that include text longer than can fit into the specified number of SMS.

E-Mail Access Restrictions

The previous section described the basic settings for e-mail to SMS or MMS. However, it did not discuss access restrictions. Access restrictions are extremely important, as you would not want to inadvertently deploy an e-mail to SMS or MMS gateway that was open for anyone to use.

By default, NowSMS will only allow e-mail messages to be sent to SMS or MMS for user accounts that are defined in the "MMSC Users" list. *(Yes, even if you are sending e-mail to SMS, these accounts are defined in the "MMSC Users" list.)*

So, if you only want to allow e-mail messages to be sent to a few addresses, you can define those phone numbers in the **"MMSC Users"** list. NowSMS then allows you to send to either phonenumber@mms.or.sms.domain.name or alias@mms.or.sms.domain.name.

In this default configuration, NowSMS will accept messages for any address defined in the "MMSC Users" list, but will reject attempts to send to any phone numbers that are not defined in that list.

What if you have bulk sending needs, and want to be able to send to any phone number?

Well, the first consideration is that you want to limit who can do this, so that you do not run up your SMS bill.

NowSMS supports three approaches for allowing for sending e-mail to any phone number via SMS or MMS.

1.) The first approach uses SMTP AUTH *(SMTP Authentication)*. Accounts can be defined in the "SMS Users" list. If SMTP login capability is enabled for the "SMS Users" account, an SMTP e-mail client can be configured to connect to NowSMS and authenticate itself with a defined username and password.

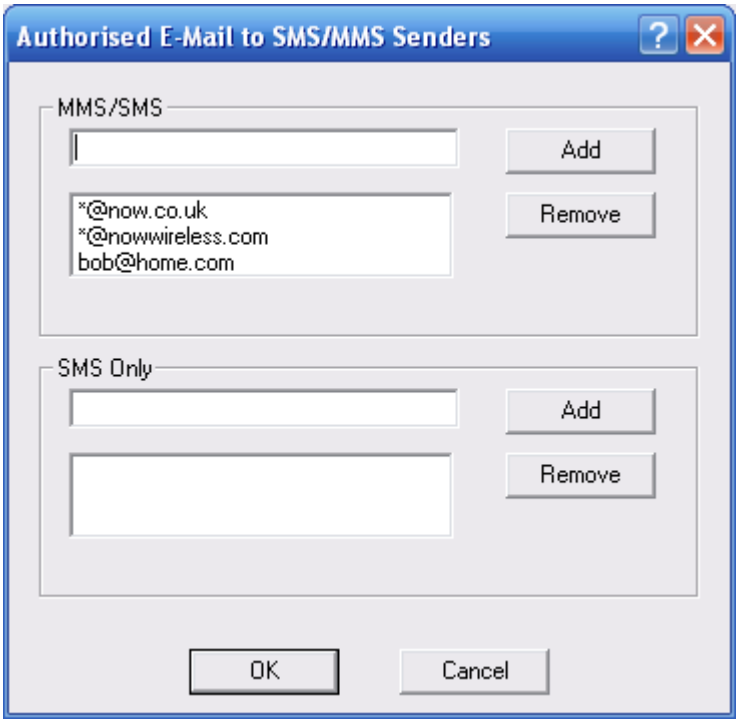
If the SMTP e-mail client authenticates with a defined username and password, NowSMS will allow the client to send to any phonenumber@mms.or.sms.domain.name. *(If you*

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

would like the account to be able to only send to SMS recipients, set "Web Menu Options" = "Text SMS Only" when configuring the "SMS Users" account.)

In this mode, the SMTP E-Mail client can also be configured to connect to the NowSMS POP3 server to receive messages. SMTP is the protocol the e-mail protocol that is used when an e-mail client wants to send a message. POP3 is one of the e-mail protocols that can be used when an e-mail client wants to receive messages from a server. When configuring POP3/SMTP e-mail client, the SMTP server is often referred to as the "Outbound Mail Server", while the POP3 server is referred to as the "Inbound Mail Server". The NowSMS POP3 Server will be discussed in more detail under the heading **Using the POP3 Server** later in this section.

2.) The second approach is to define specific e-mail addresses that are allowed to send SMS or MMS messages via NowSMS without requiring authentication. The **"Authorised E-Mail to SMS/MMS Senders"** button on the "MMSC" configuration dialog displays a dialog where these e-mail addresses can be defined.



If the SMTP server receives an e-mail from an address that is listed in the "MMS/SMS" list, it will allow that e-mail address to send an MMS or SMS message to any phone number. If the SMTP server receives an e-mail from an address that is listed in the "SMS Only" list, it will allow that e-mail address to send an SMS message to any phone number.

The authorised sender list can include e-mail addresses which include wildcards of "*" or "?". "*" matches zero or more characters. "?" matches exactly one character. These wildcards are often used to indicate that all e-mail addresses from a particular domain are to be considered authorised *(e.g., *@domain.com)*.

Use caution when deploying a server that authenticates based upon the authorised senders list. It is very easy to forge sender e-mail addresses when using internet e-mail protocols.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

For this reason, most customers who use this authorised sender list will want to make sure that the SMTP port of the NowSMS server is only accessible over their local network.

3.) The third approach involves defining "trusted" IP addresses that are allowed to submit SMS or MMS messages to NowSMS via SMTP without requiring authentication. This might be the IP address of particular workstations, or it might be the IP address of an internal corporate mail server that is configured to relay mail to NowSMS.

To define trusted IP addresses which are allowed to submit e-mail messages to any phone number, you must manually edit the MMSC.INI file. Under the [MMSC] section header, you can add a parameter:

```
SMTPIPAddressList=ip.address.1,ip.address.2,192.168.1.*
```

This parameter can have a comma-delimited list of IP addresses *(no white space, if you specify multiple addresses, separate with a comma only)* that are allowed to connect via SMTP and send to *any* phone number using an addressing format of phonenumber@mms.or.sms.domain.name. The IP addresses specified, can also include a "*" place holder as a wildcard match for all addresses in that network *(for example, *.*.*.* would open access to any address, so you could use that value if you have complete confidence in your firewall)*.

Additional E-Mail Settings

Earlier in this section, we skipped over the "Require AUTH" setting on the "MMSC" configuration dialog. This setting allows you to also require authentication before being able to send to any phone numbers defined in the "MMSC Users" list.

As discussed in the previous section, in a default configuration, NowSMS will accept messages for any address defined in the "MMSC Users" list, but will reject attempts to send to any phone numbers that are not defined in that list.

If **"Require AUTH"** is checked, even if a user account is defined in the "MMSC Users" list, NowSMS will only route SMS or MMS for that phone number if the submitting SMTP client authenticates via SMTP AUTH with a valid account, or the sending e-mail address is in the "authorised sender" list, or the submitting IP address is defined in the "SMTPIPAddressList" setting.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Using the POP3 Server

The POP3 server allows user accounts defined in the "SMS Users" dialog to connect via the POP3 e-mail protocol in order to receive SMS or MMS messages. When defining an account in the "SMS Users" dialog, it is necessary to check "Enable SMTP Login for this user" before the user account will be allowed to connect using either POP3 or SMTP.

To understand how the POP3 server can be used, it is important to understand the difference between POP3 and SMTP.

SMTP is the protocol the e-mail protocol that is used when an e-mail client wants to send a message.

POP3 is one of the e-mail protocols that can be used when an e-mail client wants to receive messages from a server.

When configuring POP3/SMTP e-mail client, the SMTP server is often referred to as the "Outbound Mail Server", while the POP3 server is referred to as the "Inbound Mail Server".

To define an e-mail account to an e-mail client that supports POP3/SMTP, it is necessary to define a connection to both the POP3 and SMTP server

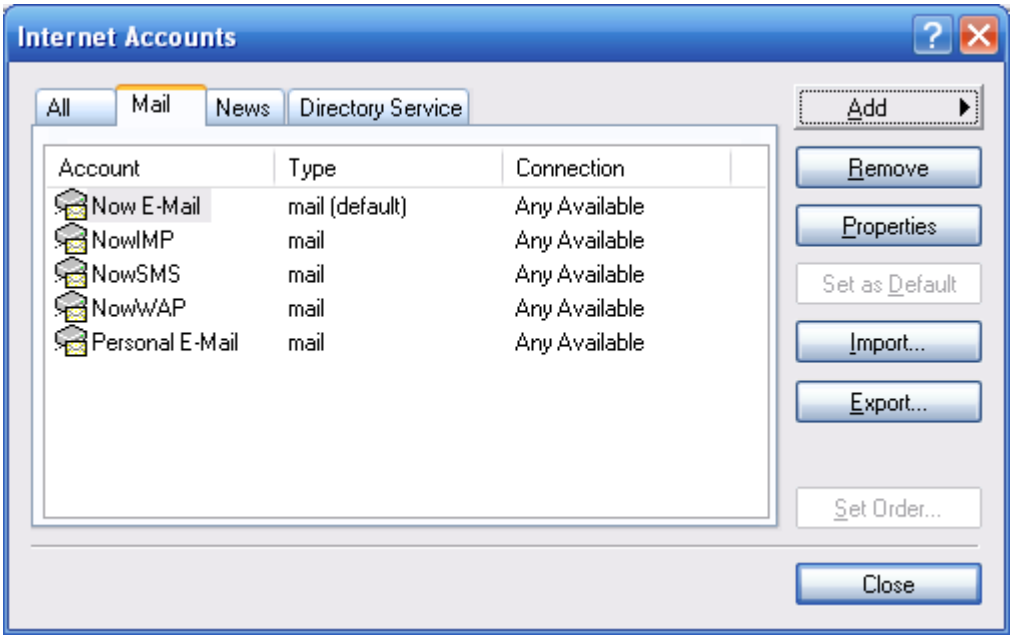
In the following sections we will detail the process of defining this type of e-mail account to either Outlook Express or Mozilla Thunderbird.

While it is possible to configure your e-mail client to connect to multiple e-mail accounts, so that you can connect to your normal e-mail server and NowSMS simultaneously, it may be easier to configure a separate e-mail client for NowSMS. The choice is yours, but especially if you are configuring the system for novice users, it may help them to think that they use a different e-mail client whenever they want to send SMS or MMS messages.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Using POP3/SMTP with Outlook Express

Select Tools/Accounts from the menu bar, and Outlook Express will display the e-mail accounts already defined.



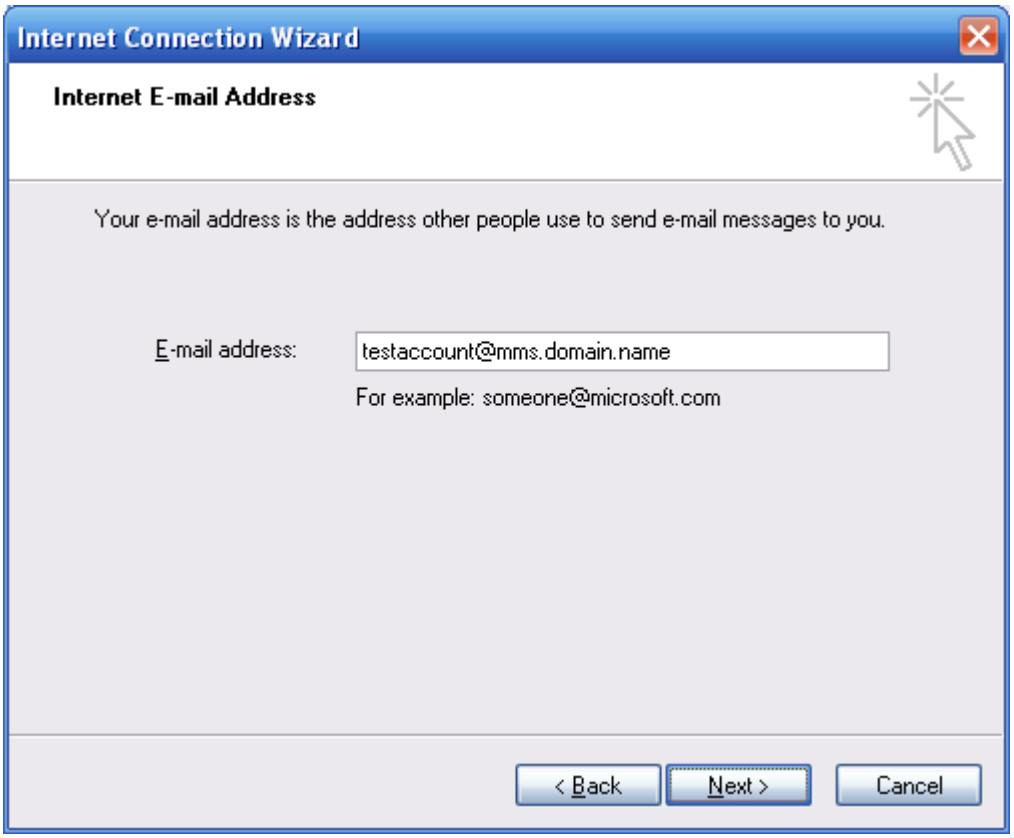
To add a new e-mail account for connecting to an account on the NowSMS server, select "Add" / "Mail", and Outlook Express will start its "Internet Connection Wizard".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The "Display Name" field is not significant to NowSMS, but will be displayed to identify the account within Outlook Express.

Press "Next".

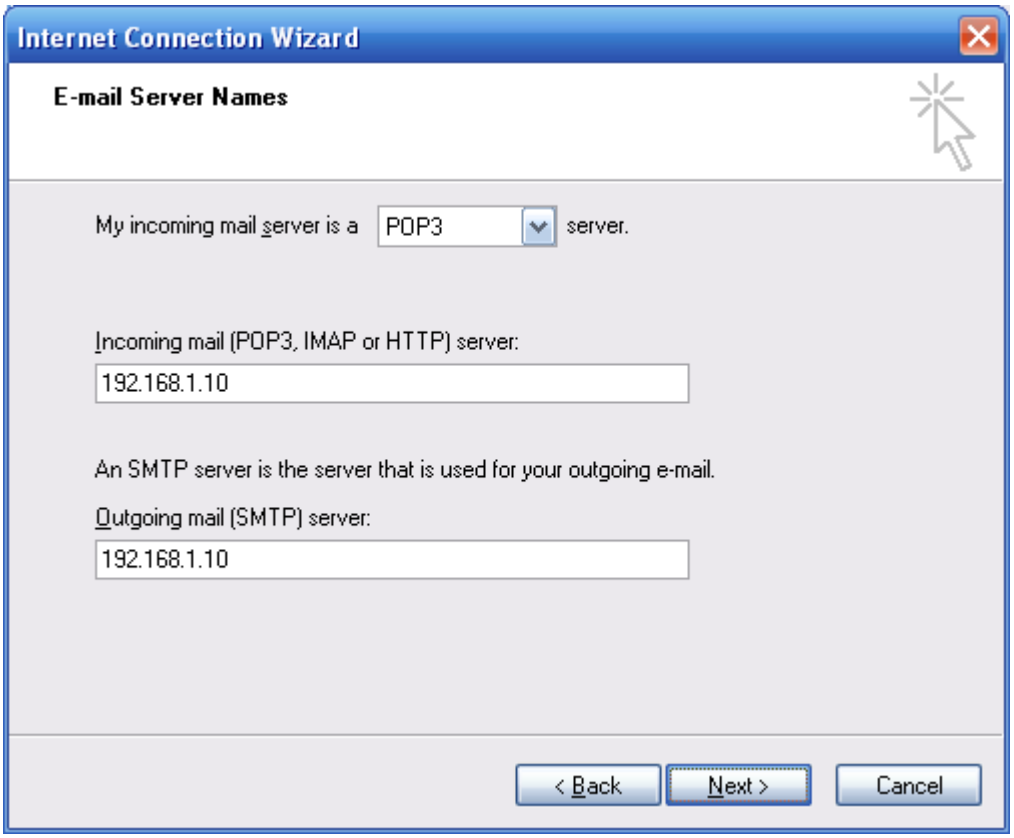
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



The "E-Mail Address" for the account is the "User Name" defined for the "SMS Users" account @ "Domain Name for MMS E-Mail" as configured on the "MMSC" page of the configuration dialog.

Press "Next".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



Select "POP3" as the incoming mail server type.

Outlook Express then wants to know the address of the "Incoming" and "Outgoing" mail servers. Input the IP address or DNS host name of the NowSMS server for both the "Incoming" and "Outgoing" mail servers.

Note that Outlook Express assumes that you are using the standard TCP/IP port numbers for SMTP (25) and POP3 (110). If you are using non-standard port numbers, you will need to perform some advanced configuration steps after completing this wizard.

Press "Next".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"Account Name" and "Password" should be configured with the "User Name" and "Password" from the account defined in the NowSMS "SMS Users" list.

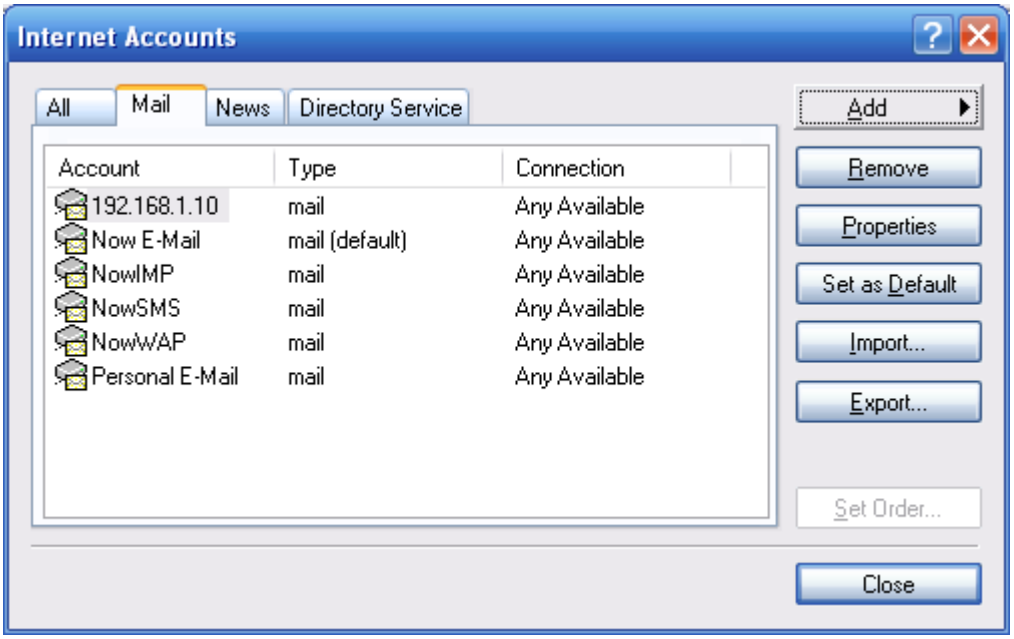
The NowSMS POP3 Server does not support Microsoft's Secure Password Authentication protocol. *(It does support secure login via APOP, which is supported by some other e-mail clients.)*

Press "Next".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The Internet Connection Wizard indicates that setup is complete after pressing "Finish", and you are returned to the list of e-mail accounts defined to Outlook Express.

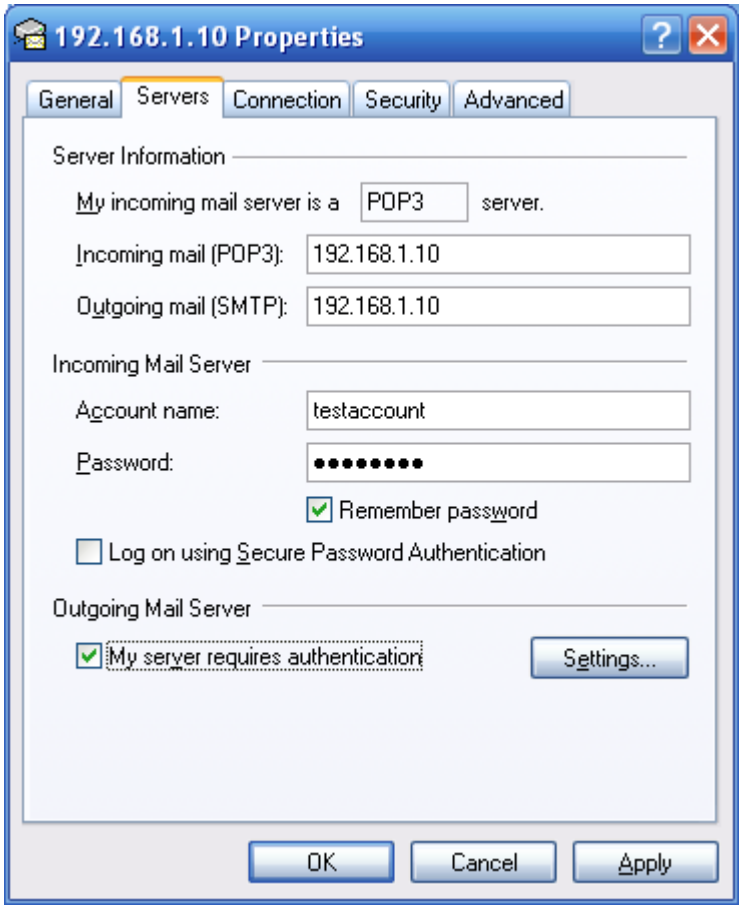
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



The IP address or host name will be displayed as the account description for the account that was just defined. It is now necessary to edit the properties for that account to enable some advanced settings. Highlight the account and press the "**Properties**" button.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

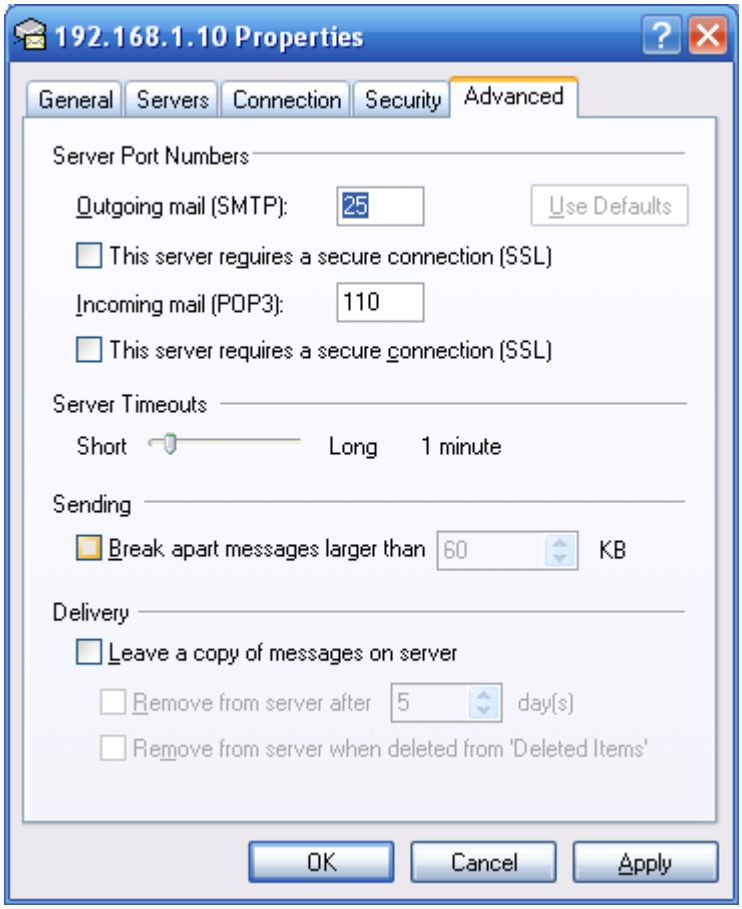
Select the "**Servers**" page in the account properties dialog.



Check "**My server requires authentication**". This setting tells NowSMS that it must send a user name and password whenever it sends an e-mail message using this account. This is known as "**SMTP Authentication**", and this is how NowSMS can identify that the user is allowed to send SMS/MMS messages.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If you are using non-standard TCP/IP ports for the SMTP or POP3 server, it is also necessary to select the "Advanced" account properties dialog.

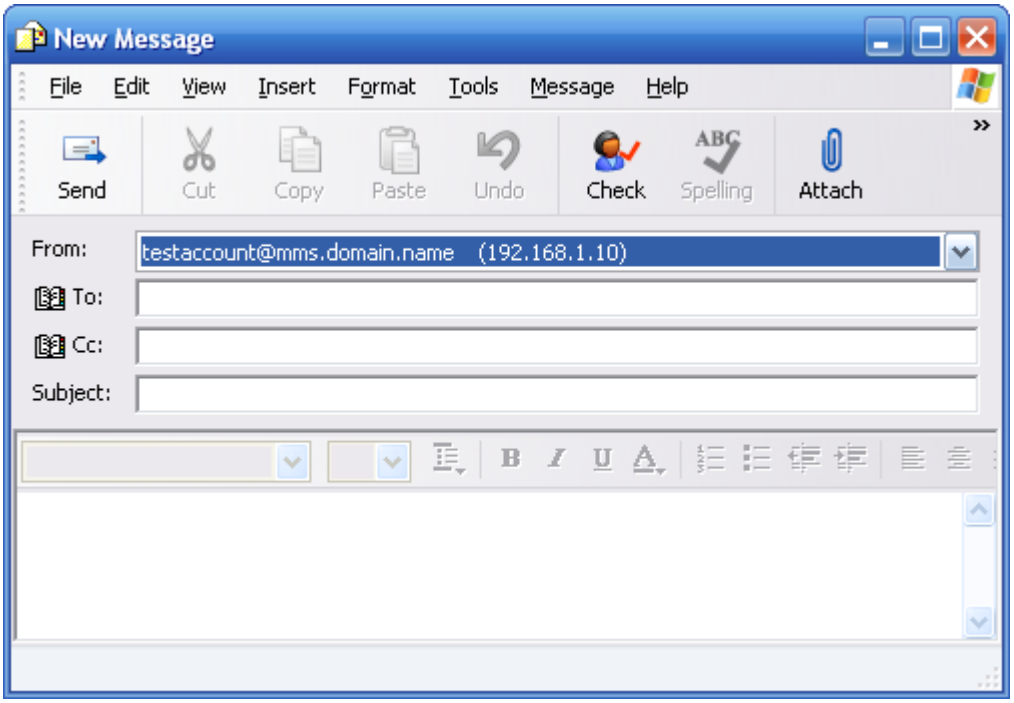


Here you can specify the appropriate ports for the SMTP and POP3 servers.

Press **OK** to complete the configuration.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If multiple e-mail accounts are configured in Outlook Express, select this e-mail account in the "From:" field whenever you want to use the e-mail client to send an SMS/MMS message via NowSMS:

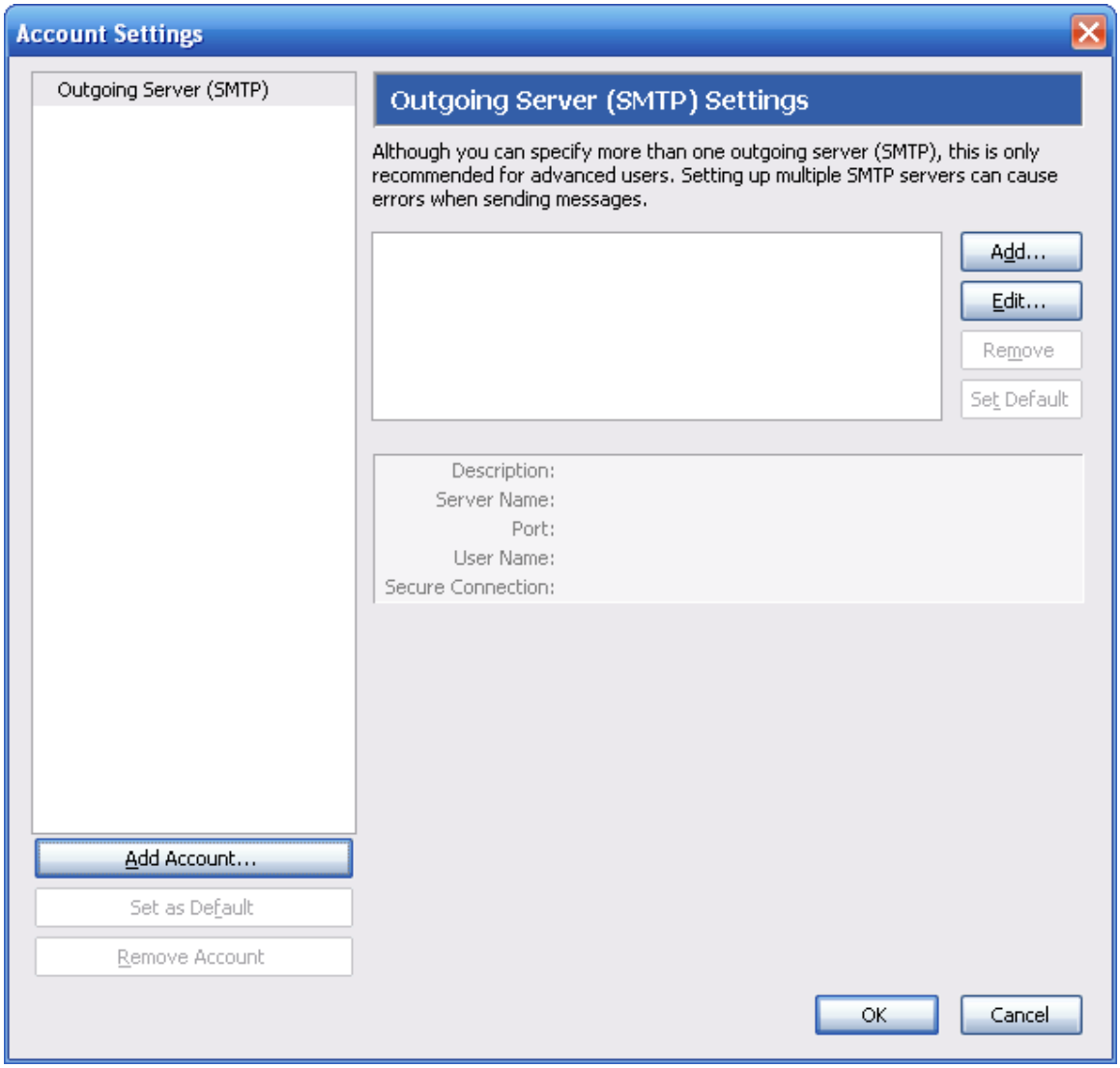


To send an SMS or MMS message via the e-mail client, in the "To" field, enter phonenumber@mms.domain.name or phonenumber@sms.domain.name, where the domain name matches either the "Domain Name for MMS E-Mail" or "Domain Name for SMS E-Mail" as defined on the "MMSC" configuration dialog.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

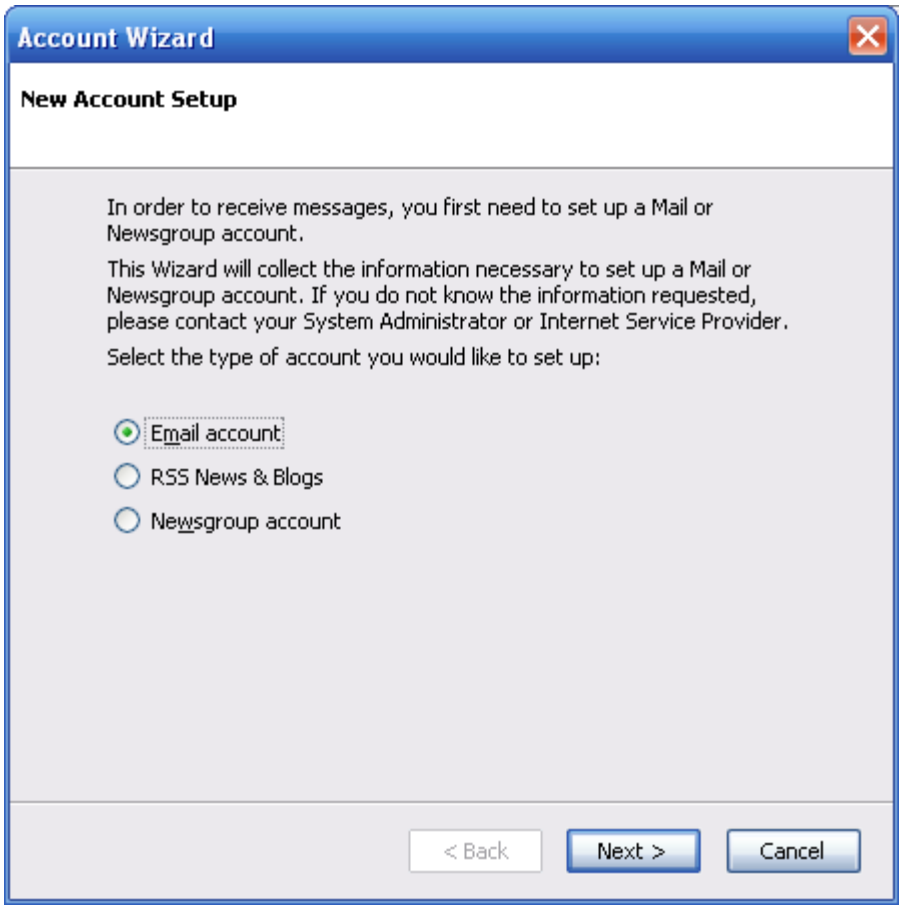
Using POP3/SMTP with Mozilla Thunderbird

Select Tools/Account Settings from the menu bar, and Thunderbird will display the "Account Settings" setup.



Select **Add Account** to define a new e-mail account.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



Select **Email account** as the account type.

Press "Next".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

"Your Name" should be a descriptive full name associated with the account.

The "Email Address" for the account is the "User Name" defined for the "SMS Users" account @ "Domain Name for MMS E-Mail" as configured on the "MMSC" page of the configuration dialog.

Press "Next".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

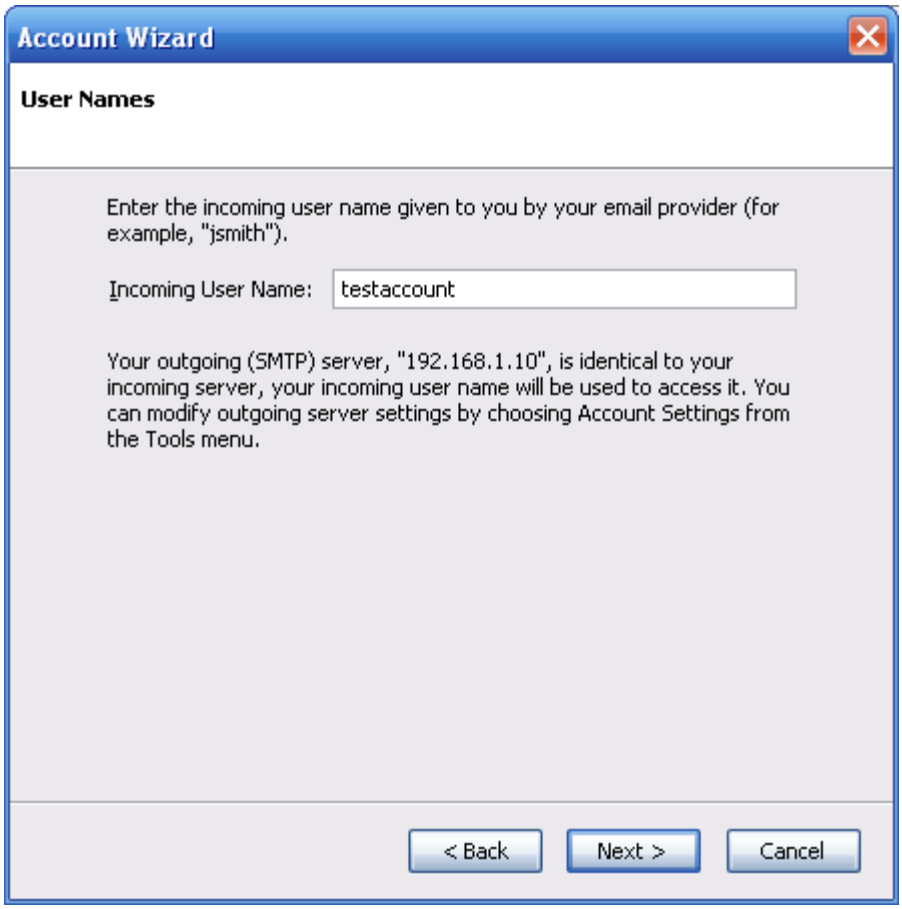
Select "POP3" as the incoming server type.

Define the IP address or DNS host name of the NowSMS server for both the "Incoming Server" and "Outgoing Server".

Note that Thunderbird assumes that you are using the standard TCP/IP port numbers for SMTP (25) and POP3 (110). If you are using non-standard port numbers, you will need to perform some advanced configuration steps after completing this wizard.

Press "Next".

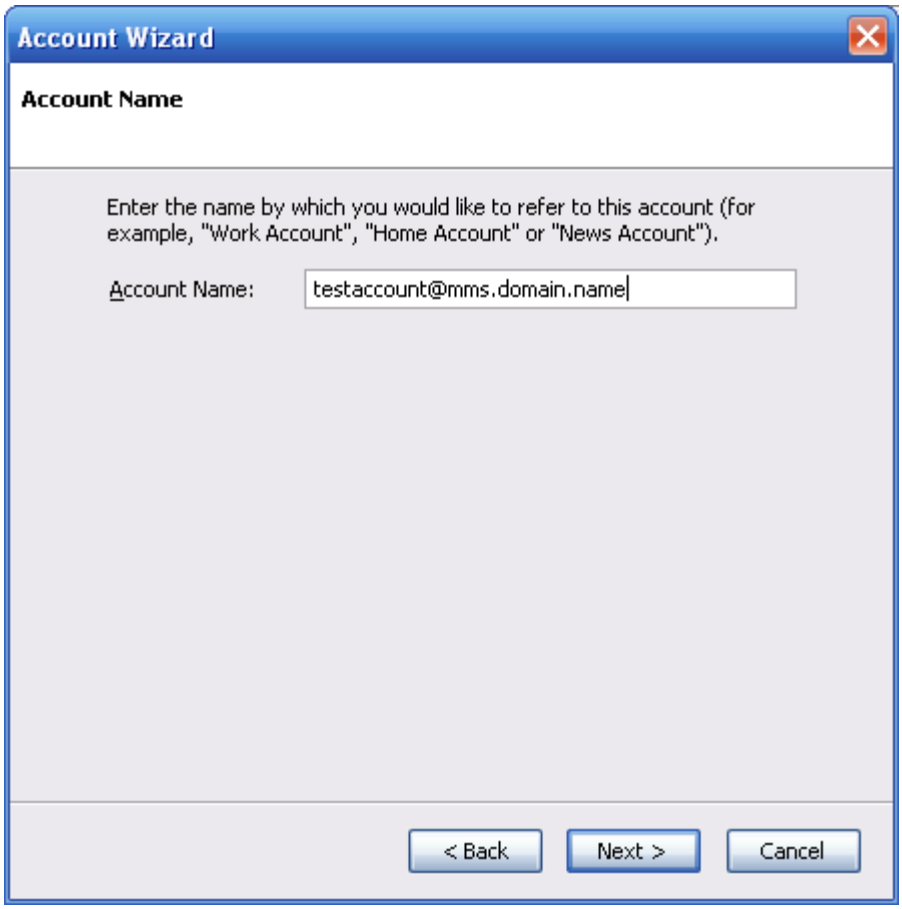
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



"Incoming User Name" is the "User Name" defined for the account in the "SMS Users" list of NowSMS.

Press "Next".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



"Account Name" is a descriptive name for the account, which defaults to the e-mail address. This value is only used within Thunderbird.

Press "Next".

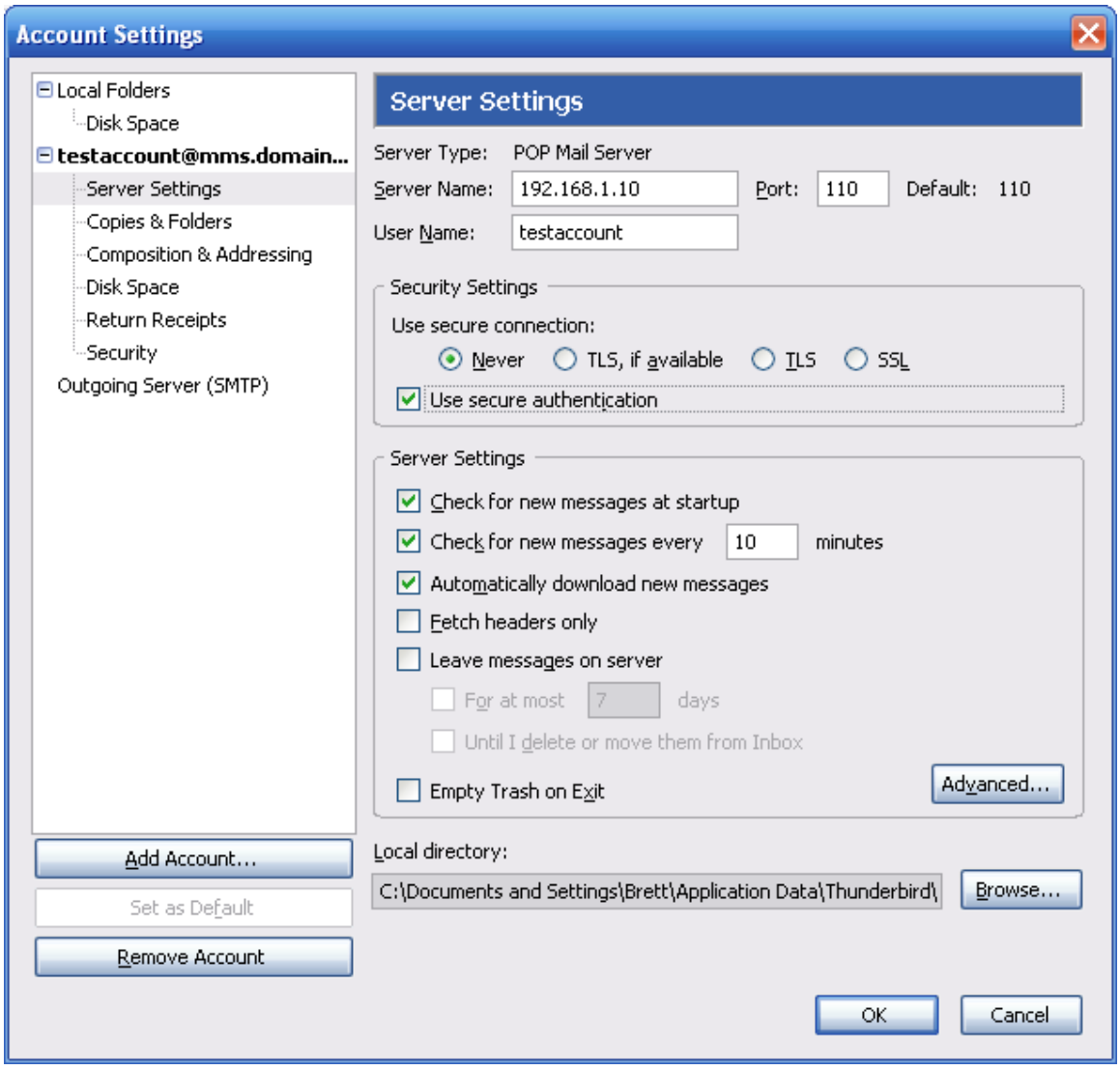
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Thunderbird displays a summary of the account settings. Use the "Back" button to correct a setting, or "Finish" to save the settings.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If you are using a non-standard TCP/IP port for the POP3 server *(other than 110)*, it is necessary to edit the Thunderbird account settings for this server connection.

This is performed in the "Server Settings" area of the "Account Settings" configuration, as shown below.

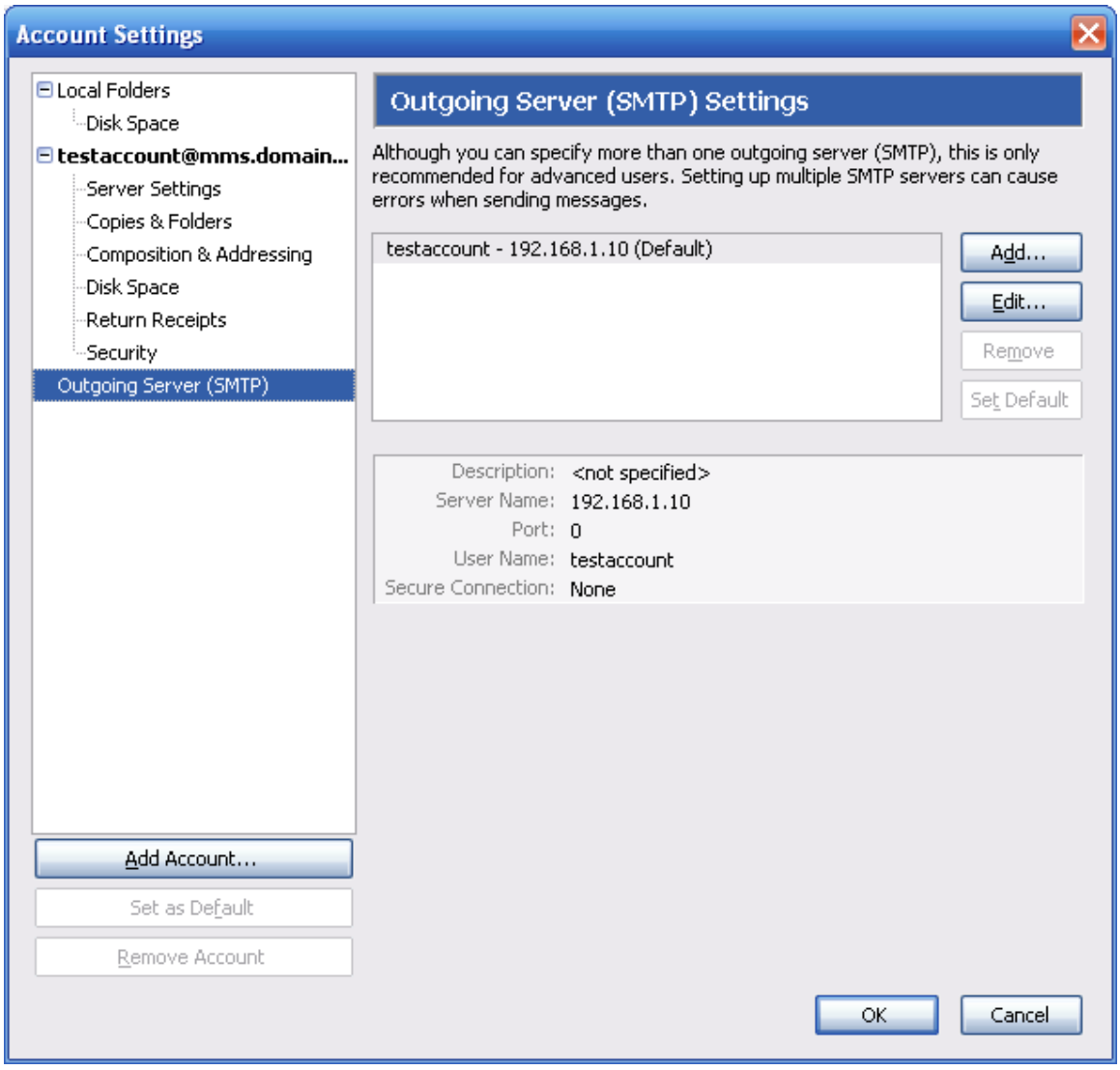


It is also possible to check the "Use secure authentication" button to enable the use of APOP for POP3 authentication.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

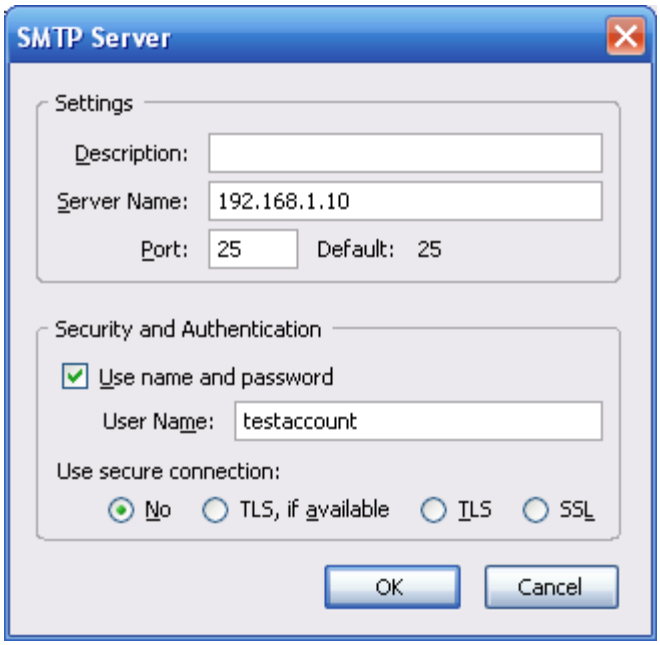
If you are using a non-standard TCP/IP port for SMTP *(other than 25)*, it is necessary necessary to edit the Thunderbird account settings for this server connection.

This is performed in the "Outgoing Server (SMTP)" area of the "Account Settings" configuration, as shown below.



Select "Edit" to edit the settings for the SMTP server, including the SMTP port number.

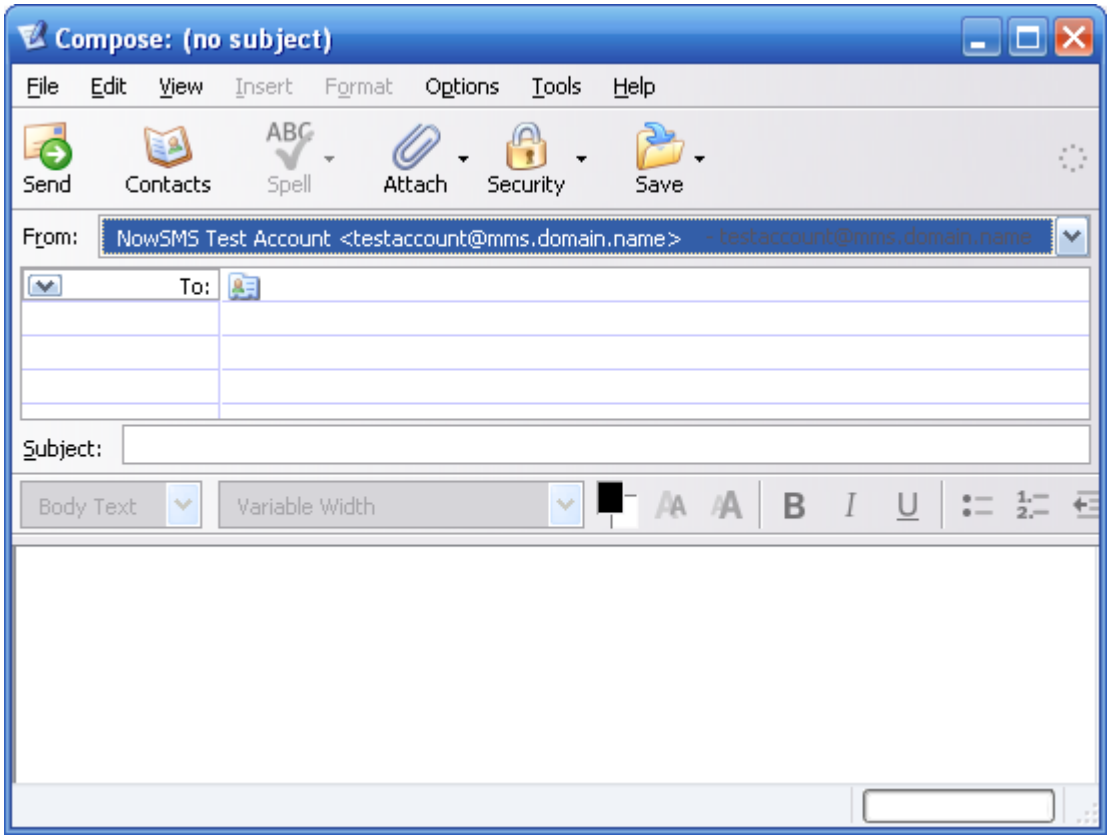
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



Press **OK** (several times if necessary) to exit the configuration.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If multiple e-mail accounts are configured in Thunderbird, select this e-mail account in the "From:" field whenever you want to use the e-mail client to send an SMS/MMS message via NowSMS:



To send an SMS or MMS message via the e-mail client, in the "To" field, enter phonenumber@mms.domain.name or phonenumber@sms.domain.name, where the domain name matches either the "Domain Name for MMS E-Mail" or "Domain Name for SMS E-Mail" as defined on the "MMSC" configuration dialog.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Receiving SMS or MMS Messages via POP3

The previous sections explained how to configure an e-mail client to connect to the NowSMS server using POP3 and SMTP. As part of that explanation, it was shown how an e-mail client can submit an SMS or MMS message using SMTP.

However, what is not obvious from this explanation is how the NowSMS server converts SMS or MMS messages for delivery to an e-mail client via POP3.

Routing SMS Messages to a POP3 Client

One of the limitations of using the e-mail to SMS facilities is that it is not possible to support SMS to e-mail replies for multiple users.

The e-mail to SMS facility can allow any e-mail sender to send an SMS message to any phone number. However, if the phone recipient attempts to reply to the SMS message, when the reply message is received by NowSMS, it is not possible to reliably match the reply to the originally sent message. This is because there is no attribute that can be set in the SMS message that indicates that it is a reply.

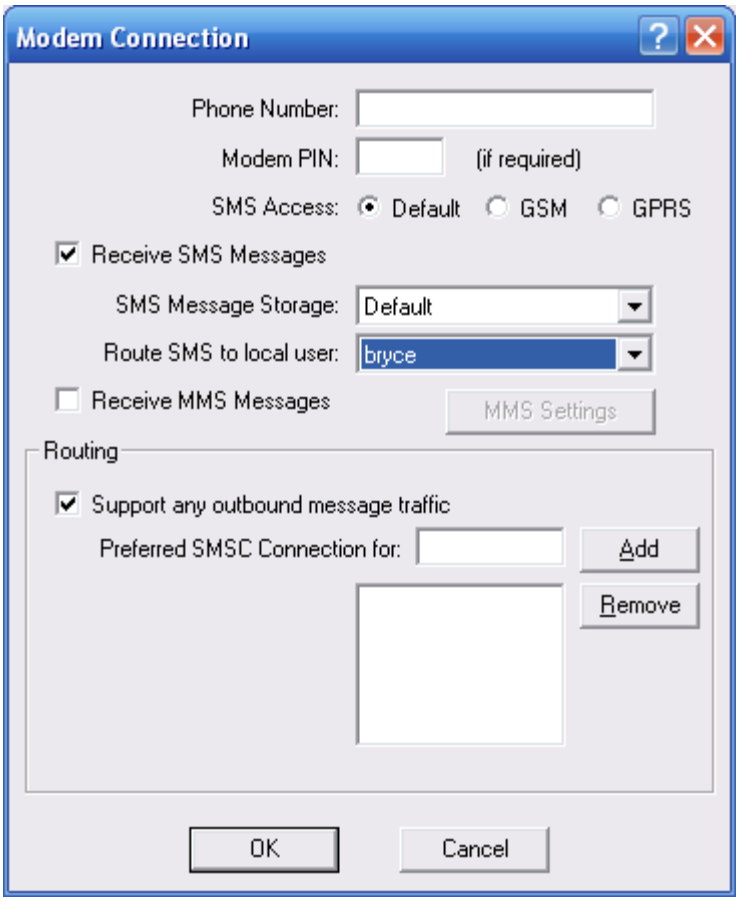
Therefore, it is only possible to define static routes for SMS to e-mail delivery.

If multiple e-mail users need to be able to send SMS messages and receive replies SMS replies back to e-mail, then it is necessary to dedicate a phone number (or GSM modem) for each of these users.

There are three ways to configure NowSMS to route received SMS messages to an "SMS Users" account associated with a POP3 e-mail client.

1.) All SMS messages received via a specific GSM modem connection can be routed to a specific "SMS Users" account. To configure this support, highlight the GSM modem definition on the "SMSC" configuration dialog, and press the "Properties" button.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



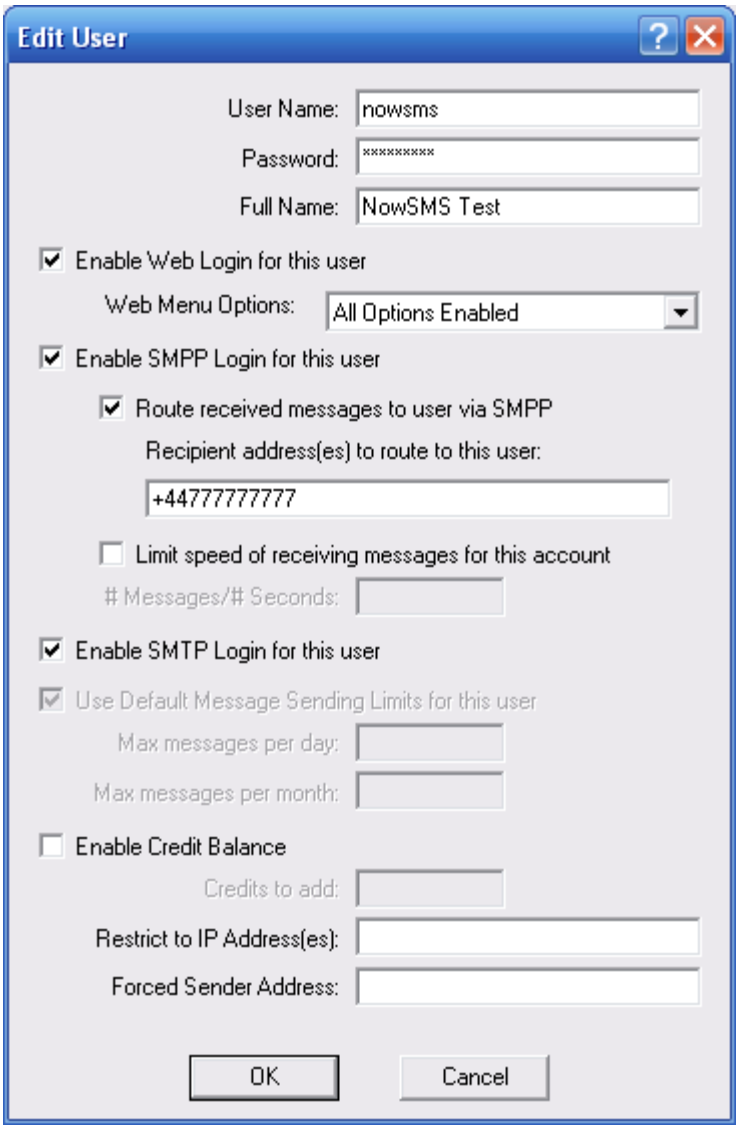
Use the "Route SMS to local user" parameter to specify an account which should receive all SMS messages that are received via this GSM modem.

2.) The "2-way" facility in NowSMS, which is used to process received SMS messages, is normally used to deliver received SMS messages to an HTTP script or executable program. However, it can also be used to route received SMS messages to an e-mail address.

When defining a "2-way" command in the "Received SMS Command Table", it is possible to define a "mailto:" link in the "Command to Execute" field. This facility can be used to route a message to a local "SMS Users" account by specifying a "Command to Execute" of "mailto:localuser@mms.domain.name", where "localuser" is the name of the "SMS Users" account, and "mms.domain.name" is the "Domain Name for MMS E-Mail" as defined in the "MMSC" configuration dialog. For additional information, please see **2-Way SMS** on page 236.

3.) It is possible to specify that all messages received for one or more phone numbers should be routed to a specific user account. This is done by enabling features in the "SMS Users" account definition that are also used for SMPP client connections:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



To use this support, check **"Enable SMPP Login for this user"** *(even though the account will be using SMTP instead of SMPP)*. Check **"Route received messages to user via SMPP"**, and specify one or more phone numbers *(separate multiple phone numbers with a comma)*, where if the recipient of a message received by the gateway matches a phone number listed here, the message will be queued for delivery to this "SMS Users" account, where the account can retrieve it using either the SMPP or SMTP protocol.

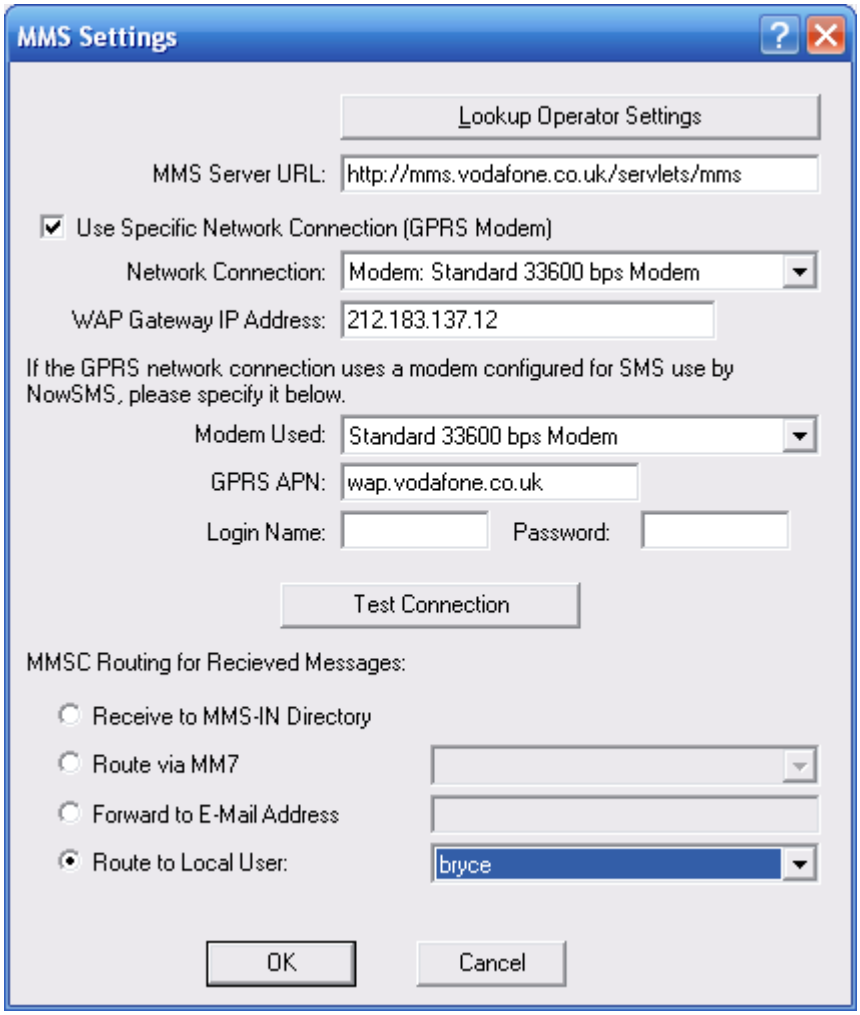
Note that this routing applies to all messages processed by NowSMS where the recipient phone number associated with the message matches a phone number in the "Recipient address(es) to route to this user" field. It applies to both messages received from an SMSC connection, as well as if another user account on the NowSMS server attempts to send a message to this phone number.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Routing MMS Messages to a POP3 Client

There are three ways to configure NowSMS to route received MMS messages to an "SMS Users" account associated with a POP3 e-mail client.

1.) It is possible to specify that all MMS messages received over a specific GPRS modem be routed to a local user account. When defining the "Receive MMS Settings" for a GPRS modem connection, the "Route to Local User" field can be used to specify that MMS messages received via this modem should be routed to a local user account:



2.) All messages received via any defined "MMSC VASP" connection can also be configured to be routed to a local user account, using a configuration setting similar to the one defined for GPRS modems.

3.) Unlike SMS, it is possible to configure NowSMS such that if a phone user replies to an MMS message, the reply can be directed to the original message sender. To enable this support, when defining a GPRS modem connection for sending MMS messages (see

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Connecting to an Operator MMSC - Sending MMS Messages on page 142), check the **"Embed Original Sender Address in Subject Header"** setting.

When this setting is enabled, NowSMS will automatically insert "(FM: username)" into the subject header of the MMS message.

When NowSMS receives an MMS message, it checks the subject header for this information. If the MMS message is a reply, this information should still be present in the subject header, and NowSMS can route the MMS reply to the appropriate e-mail recipient.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Routing SMS Messages to E-Mail

NowSMS can be configured to route received SMS messages to an e-mail address. This configuration is similar to the configuration of **Routing SMS Messages to a POP3 Client** defined on page 271. However, only the method of using the "2-way" command facility can be used to route received SMS messages to an e-mail address other than a local POP3 client account.

The "2-way" facility in NowSMS, which is used to process received SMS messages, is normally used to deliver received SMS messages to an HTTP script or executable program. However, it can also be used to route received SMS messages to an e-mail address.

When defining a "2-way" command in the "Received SMS Command Table", it is possible to define a "mailto:" link in the "Command to Execute" field.

Possible "Command to Execute" values for this purpose include:

"mailto:user@domain.com" - The SMS message will be put into an e-mail message and sent to user@domain.com.

"mailto:@@SMSPREFIX@@" - NowSMS will look for an e-mail address to be the first text of the received SMS message. NowSMS will send the text of the SMS message after this e-mail address to the e-mail address that appears as the first text of the SMS message. (Example: SMS message text = "nowsms@now.co.uk This is a test" -- e-mail message will be sent to "nowsms@now.co.uk" with text "This is a test".)

If "mailto:@@SMSPREFIX@@" is used, and NowSMS does not see a valid e-mail address at the start of the SMS message, NowSMS will reply with the following text: "Invalid e-mail recipient. Please specify a valid e-mail address to start the text of your message."

For additional information on defining 2-Way SMS commands, please see **2-Way SMS** on page 236.

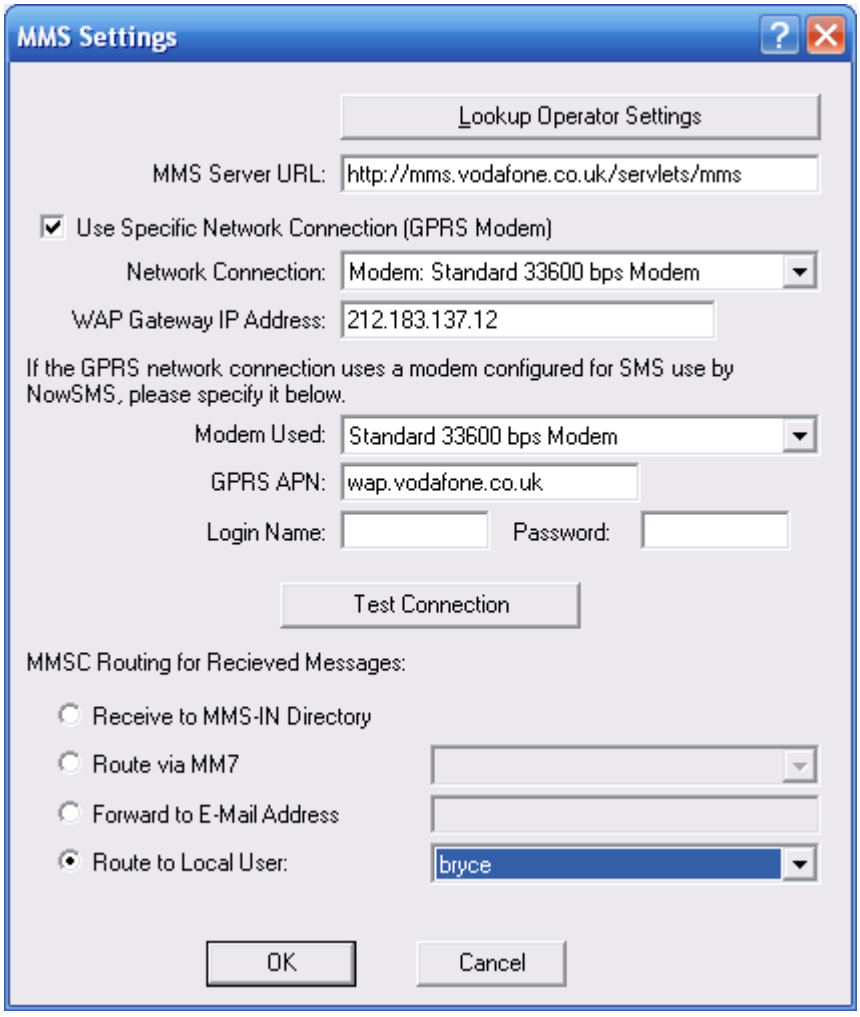
© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Routing MMS Messages to E-Mail

MMS to e-mail is part of the standard MMSC functionality when a mobile phone is configured to use NowSMS as an MMSC. An MMS client can address a message to either another phone number or an e-mail address. When NowSMS receives an MMS addressed to an e-mail address, it automatically tries to route the message via e-mail.

For configurations where NowSMS is not the MMSC, but is instead acting as a gateway to another MMSC, it is possible to configure NowSMS to route MMS messages to an e-mail address. This configuration is similar to the configuration of **Routing MMS Messages to a POP3 Client** defined on page 274.

1.) It is possible to specify that all MMS messages received over a specific GPRS modem be forwarded to a specific e-mail address. When defining the "Receive MMS Settings" for a GPRS modem connection, the "Forward to E-Mail Address" field can be used to specify that MMS messages received via this modem should be sent to the specified e-mail address:



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

2.) All messages received via any defined "MMSC VASP" connection can also be configured to be forwarded to an e-mail address, using a configuration setting similar to the one defined for GPRS modems.

3.) Unlike SMS, it is possible to configure NowSMS such that if a phone user replies to an MMS message, the reply can be directed to the original message sender. To enable this support, when defining a GPRS modem connection for sending MMS messages (see **Connecting to an Operator MMSC - Sending MMS Messages** on page 142), check the **"Embed Original Sender Address in Subject Header"** setting.

When this setting is enabled, NowSMS will automatically insert "(FM: username)" into the subject header of the MMS message.

When NowSMS receives an MMS message, it checks the subject header for this information. If the MMS message is a reply, this information should still be present in the subject header, and NowSMS can route the MMS reply to the appropriate e-mail recipient.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

SMS Gateway Billing and Charging: Accounting Callbacks

Accounting callbacks provide an interface between the NowSMS SMS Gateway and external billing and charging systems.

These accounting callbacks are HTTP-based. When accounting callbacks are enabled, NowSMS will issue HTTP requests to a customer supplied URL in order to interface with the customer billing and charging systems.

To enable SMS accounting callbacks, it is necessary to manually edit the SMSGW.INI configuration file, and define the callback URL under the [SMSGW] section header, using the following configuration parameter:

```
SMSAccountingURL=http://server/path
```

Whenever the SMS Gateway processes an SMS message, it issues an accounting callback by issuing an HTTP transaction to the callback URL. Variables describing the SMS message transaction are appended to the SMSAccoutingURL as HTTP GET CGI-style variables, with standard URL escaping applied for encoding reserved characters.

For example:

```
http://server.name/path?PreAuth=Yes&Type=SMSSend&From=UserAccount&To=
%2B447777777777&MsgCount=1&SubmitIP=127.0.0.1&Text=This%20is%20a%20test.
```

(These variables and transaction types will be described later in this section.)

Some accounting callbacks are informational only, and exist to record charging information after the SMS Gateway has processed a message.

However, there is also a pre-authorisation callback which occur before NowSMS processes a message. These pre-authorisation callbacks exist to allow the customer billing system to decide whether or not the message should be allowed. In this scenario, the callback could check available credit and reject an SMS message transaction before it is accepted by NowSMS.

The remainder of this section provides additional details on the parameters supported by these callbacks.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

SMSSend PreAuth Callback

This callback is executed when an SMS (web, SMPP, SMTP) user is requesting to send a message.

This is a “pre-authorisation” request, and does not mean that the message will actually be accepted by NowSMS for delivery. If NowSMS cannot successfully connect to the accounting URL, or the URL returns a response other than a standard “HTTP 200 OK” response, the user request to send a message will be blocked. A “PreAuth” request to send a message will also be blocked if the response includes the text “PreAuth=Deny”.

The following variables will be set for a pre-authorisation request:

PreAuth=Yes (indicates that the message is a Pre-Authorisation Request)
Type=SMSSend
From=Defined "SMS Users" Account
To=Comma delimited list of message recipients (will not be present if message is addressed to more than 100 recipients)
MsgCount=#### (number of recipients user is requesting to send the message to)
SubmitIP=a.b.c.d
SMSCRoute=xxxxxx (optional, will be present only if an explicit route was requested in message submission)
Sender=xxxxxx (optional, will be present only if a sender address was specified in message submission)
Binary=1 (optional, will be present if the message is binary)
PID=# (optional, will be present only if a non-zero PID value was specified in message submission)
DCS=# (optional, will be present only if a non-zero DCS value was specified in message submission)
UDH=HexString (optional, will be present only if message contains User Data Header)
Data=HexString (optional, will be present only if message is binary)
Text=String (optional, will be present only if message is text)
ReceiptRequested=Yes (optional, will be present only if message is requesting a delivery receipt ... only available in v2009.01.26 and later)

(Note: For SMTP message submissions, only PreAuth, Type, From, To and MsgCount variables will be present.)

Example:

```
http://server.name/path?PreAuth=Yes&Type=SMSSend&From=UserAccount&To=
%2B447777777777&MsgCount=1&SubmitIP=127.0.0.1&Text=This%20is%20a%20test.
```

Note that URL escaping is performed when building the URL string. Most HTTP scripting languages will automatically unescape these parameters for you (e.g., %2B is translated back to "+" and %20 is translated back to a space character).

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

SMSSend Accounting Callback

This callback is executed after an SMS message that has been sent by an SMS user has been accepted by the gateway for delivery.

In NowSMS 2009 and later, NowSMS will check the response to the HTTP request. If this response includes the text "SMSCRoute=xxxxx", then NowSMS will apply this SMSC route for the message. The specified route "xxxxx" can either be the name of a particular SMSC connection (e.g., "SMPP - host:port"), or it can be the value of the "RouteName=" attribute defined for one or more connections. (For more information on the "RouteName=" attribute, see http://blog.nowsms.com/2008/07/routing-sms-messages-to-specifc-smsc.html.)

The following variables will be set for the accounting callback:

Type=SMSSend
From=Defined "SMS Users" Account
To=Message Recipient Phone Number (if the message is sent to multiple recipients, this callback is repeated for each recipient)
MessageID=Message ID assigned to the message by NowSMS
SubmitIP=a.b.c.d
SMSCRoute=xxxxxx (optional, will be present only if an explicit route was requested in message submission)
Sender=xxxxxx (optional, will be present only if a sender address was specified in message submission)
Binary=1 (optional, will be present if the message is binary)
PID=# (optional, will be present only if a non-zero PID value was specified in message submission)
DCS=# (optional, will be present only if a non-zero DCS value was specified in message submission)
UDH=HexString (optional, will be present only if message contains User Data Header)
Data=HexString (optional, will be present only if message is binary)
Text=String (optional, will be present only if message is text)
ReceiptRequested=Yes (optional, will be present only if message is requesting a delivery receipt ... only available in v2009.01.26 and later)

(Note: For SMTP message submissions, only Type, From, and To variables will be present.)

Example:

```
http://server.name/path?Type=SMSSend&From=UserAccount&To=
%2B447777777777&SubmitIP=127.0.0.1&Text=This%20is%20a%20test.
```

Note that URL escaping is performed when building the URL string. Most HTTP scripting languages will automatically unescape these parameters for you (e.g., %2B is translated back to "+" and %20 is translated back to a space character).

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSC Billing and Charging: Accounting Callbacks

MMSC accounting callbacks provide an interface between the NowSMS MMSC and external billing and charging systems.

These MMSC accounting callbacks are HTTP-based. When accounting callbacks are enabled, the MMSC will issue HTTP requests to a customer supplied URL in order to interface with the customer billing and charging systems.

To enable MMSC accounting callbacks, it is necessary to manually edit the MMSC.INI configuration file, and define the callback URL under the [MMSC] section header, using the following configuration parameter:

```
MMSAccountingURL=http://server/path
```

Whenever the MMSC processes an MMS message, it issues an accounting callback by issuing an HTTP transaction to the callback URL. Variables describing the MMS transaction are appended to the MMSAccoutingURL as HTTP GET CGI-style variables, with standard URL escaping applied for encoding reserved characters.

For example:

```
http://server/path?PreAuth=Yes&Type=MMSSend&From=%2B449999999999&To=
%2B447777777777&MsgCount=1
```

(These variables and transaction types will be described later in this section.)

Most of the accounting callbacks are informational only, and exist to record charging information after the MMSC has processed a transaction.

However, there are also pre-authorisation callbacks which occur before the MMSC processes a transaction. These pre-authorisation callbacks exist to allow the customer billing system to decide whether or not the transaction should be allowed. In this scenario, the callback could check available credit and reject an MMS message transaction before it is accepted by the MMSC.

The remainder of this section provides additional details on the parameters supported by these callbacks.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSSend PreAuth Callback

This callback is executed when an MMS subscriber, Value Added Service Provider (VASP) or MMSC interconnect partner, is requesting to send a message.

This is a “pre-authorisation” request, and does not mean that the message will actually be accepted by NowSMS for delivery. If NowSMS cannot successfully connect to the accounting URL, or the URL returns a response other than a standard “HTTP 200 OK” response, the user request to send a message will be blocked. A “PreAuth” request to send a message will also be blocked if the HTTP response content includes the text “PreAuth=Deny”.

The following parameter variables may be set for the MMSSend pre-authorisation request:

PreAuth=Yes

The presence of this parameter indicates that this callback is a pre-authorisation request.

Type=MMSSend

The transaction type is MMSSend, indicating that a request is being made to send an MMS message.

From=SenderPhoneNumber

This parameter contains the phone number of the subscriber that is sending the message. Note that URL escaping rules require the "+" symbol to be encoded as "%2B".

To=RecipientPhoneNumber (may be a comma delimited list with multiple recipients)

This parameter contains one or more recipient phone numbers. If more than one phone number is present, this will be a comma delimited list of recipient phone numbers. (Note that URL escaping rules require the "," symbol to be encoded as "%2C".)

VASPIN=MmscVaspName

This parameter is present if the message is arriving from a Value Added Service Provider or MMSC interconnect partner. The value of this parameter refers to the account name as defined in the "MMSC VASP" list.

Note that some versions of NowSMS may preface the MmscVaspName with the text "VASP:".

VASP=MmscOutboundRoute (may be a comma delimited list if multiple recipients)

This parameter is present if the MMSC has determined that the message must be routed via an external route for delivery. The value of this parameter refers to the account name as defined in the "MMSC Routing" list.

If the message is being sent to multiple recipients, this field may contain a comma delimited list of routes with a route listed for each recipient. If there is a mix of local and remote recipients, local recipients will have a blank entry within the comma delimited list of routes.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MsgCount=*####*

This parameter specifies the number of recipients for this MMS message transaction.

(Note that the MMSSend PreAuth callback is issued only once if an MMS message is being sent to multiple recipients. The MMSSend Charging callback, which records billing and charging information after the MMSC has accepted the message, issues a separate callback for each recipient.)

Size=*####*

This parameter specifies the size of the MMS message in bytes.

Note that the MMS message size refers to the size of the data using the encoding of the protocol through which the message is being received (e.g., MM1, MM4, MM7). The actual size of the delivered MMS message may be different because of conversion between these protocols, and/or MMS header manipulation.

Also note that this parameter may not be present for all protocols. In particular, this parameter is not present for messages received via the MM4 (MMSC interconnect) protocol.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSSend Charging Callback

This callback is executed when an MMS subscriber, Value Added Service Provider (VASP) or MMSC interconnect partner, has submitted a message to the MMSC, and the MMSC has accepted the message for further processing.

NowSMS will ignore any HTTP response returned by the callback, however we recommend returning an "HTTP 200 OK" response for future compatibility reasons.

The following parameter variables may be set for the MMSSend Charging Callback:

Type=MMSSend

The transaction type is MMSSend, indicating that an MMS message has been submitted.

From=SenderPhoneNumber

This parameter contains the phone number of the subscriber that is sending the message. Note that URL escaping rules require the "+" symbol to be encoded as "%2B".

To=RecipientPhoneNumber

This parameter contains a single recipient phone number. If the original message was submitted to more than one recipient, a separate charging callback will occur for each recipient.

VASPIN=MmscVaspName

This parameter is present if the message arrived from a Value Added Service Provider or MMSC interconnect partner. The value of this parameter refers to the account name as defined in the "MMSC VASP" list.

Note that some versions of NowSMS may preface the MmscVaspName with the text "VASP:".

VASP=MmscOutboundRoute

This parameter is present if the MMSC has determined that the message must be routed via an external route for delivery. The value of this parameter refers to the account name as defined in the "MMSC Routing" list.

Note that some versions of NowSMS may preface the MmscOutboundRoute with the text "VASP:".

MessageID=assignedMessageID

This parameter records the MMS message ID assigned by MMSC. Note that if the message was sent to multiple recipients, each recipient instance shares the same message ID.

Size=####

This parameter specifies the size of the MMS message in bytes.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Note that MMS message size may differ based upon the encoding protocol (e.g., MM1, MM4, MM7). The actual size of the delivered MMS message may be different because of conversion between these protocols, and/or MMS header manipulation.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSRetrieve Accounting Callback

This callback is executed when an MMS subscriber connects to the MMSC to retrieve the content of an MMS message.

NowSMS will ignore any HTTP response returned by the callback, however we recommend returning an "HTTP 200 OK" response for future compatibility reasons.

The following parameter variables may be set for the MMSRetrieve Accounting Callback:

Type=MMSRetrieve

The transaction type is MMSRetrieve, indicating that an MMS subscriber has connected to the MMSC to retrieve the content of an MMS message.

From=SenderPhoneNumber (or EMailAddress)

This parameter contains the phone number or e-mail address of the message sender. Note that URL escaping rules require the "+" symbol to be encoded as "%2B".

To=RecipientPhoneNumber

This parameter contains the recipient phone number that is retrieving this MMS message.

MessageID=assignedMessageID

This parameter contains the MMS message ID assigned by MMSC. Note that if the message was sent to multiple recipients, each recipient instance shares the same message ID.

Size=*####*

This parameter specifies the size of the MMS message in bytes.

Note that MMS message size may differ based upon the encoding protocol (e.g., MM1, MM4, MM7). The actual size of the delivered MMS message may be different because of conversion between these protocols, and/or MMS header manipulation.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSOut Accounting Callback

This callback is executed when an MMS message is routed to an external route (VASP or MMSC interconnect) defined in the "MMSC Routing" list.

NowSMS will ignore any HTTP response returned by the callback, however we recommend returning an "HTTP 200 OK" response for future compatibility reasons.

The following parameter variables may be set for the MMSOut Accounting Callback:

Type=MMSOut

The transaction type is MMSOut, indicating that an MMS message has been routed to an external route.

From=SenderPhoneNumber (or EMailAddress)

This parameter contains the phone number or e-mail address of the message sender. Note that URL escaping rules require the "+" symbol to be encoded as "%2B".

To=RecipientPhoneNumber

This parameter contains the recipient phone number for the MMS message.

MessageID=assignedMessageID

This parameter contains the MMS message ID assigned by MMSC. Note that if the message was sent to multiple recipients, each recipient instance shares the same message ID.

Size=####

This parameter specifies the size of the MMS message in bytes.

Note that MMS message size may differ based upon the encoding protocol (e.g., MM1, MM4, MM7). The actual size of the delivered MMS message may be different because of conversion between these protocols, and/or MMS header manipulation.

VASP=MmscOutboundRoute

This parameter specifies the MMSC outbound route via which the MMS message was routed.

Note that some versions of NowSMS may preface the MmscOutboundRoute with the text "VASP:".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSOutFailed Accounting Callback

This callback is executed when an attempt is made to route an MMS message to an external route (VASP or MMSC interconnect) defined in the "MMSC Routing" list, but the attempt fails.

Information about the failure is not currently provided by this callback, but can be found in the MMSC-yyyymmdd.LOG file.

NowSMS will ignore any HTTP response returned by the callback, however we recommend returning an "HTTP 200 OK" response for future compatibility reasons.

The following parameter variables may be set for the MMSOutFailed Accounting Callback:

Type=MMSOutFailed

The transaction type is MMSOutFailed, indicating that an attempt was made to route an MMS message to an external route, but the attempt failed.

From=SenderPhoneNumber (or EMailAddress)

This parameter contains the phone number or e-mail address of the message sender. Note that URL escaping rules require the "+" symbol to be encoded as "%2B".

To=RecipientPhoneNumber

This parameter contains the recipient phone number for the MMS message.

MessageID=assignedMessageID

This parameter contains the MMS message ID assigned by MMSC. Note that if the message was sent to multiple recipients, each recipient instance shares the same message ID.

Size=####

This parameter specifies the size of the MMS message in bytes.

Note that MMS message size may differ based upon the encoding protocol (e.g., MM1, MM4, MM7). The actual size of the delivered MMS message may be different because of conversion between these protocols, and/or MMS header manipulation.

VASP=MmscOutboundRoute

This parameter specifies the MMSC outbound route via which the MMS message was routed.

Note that some versions of NowSMS may preface the MmscOutboundRoute with the text "VASP:".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSDeliveryReport PreAuth Callback

This callback is executed when a Value Added Service Provider (VASP) or MMSC interconnect partner is requesting to send a delivery report. This callback is also generated when the MMSC wants to generate a delivery a report on behalf of a local subscriber.

This is a “pre-authorisation” request, and does not mean that the delivery report will actually be accepted by NowSMS for delivery. If NowSMS cannot successfully connect to the accounting URL, or the URL returns a response other than a standard “HTTP 200 OK” response, the user request to send a message will be blocked. A “PreAuth” request to send a message will also be blocked if the HTTP response content includes the text “PreAuth=Deny”.

The following parameter variables may be set for the MMSDeliveryReport pre-authorisation request:

PreAuth=Yes

The presence of this parameter indicates that this callback is a pre-authorisation request.

Type=MMSDeliveryReport

The transaction type is MMSDeliveryReport, indicating that a request is being made to send an MMS delivery report.

From=SenderPhoneNumber

This parameter contains the phone number of the subscriber for which the delivery report is being generated (i.e., the original recipient of the message).

To=RecipientPhoneNumber

This parameter contains the phone number to which this delivery report is being sent (i.e., the original sender of the message).

VASP=MmscOutboundRoute

This parameter is present if the MMSC has determined that the delivery report must be routed via an external route for delivery. The value of this parameter refers to the account name as defined in the "MMSC Routing" list.

Note that some versions of NowSMS may preface the MmscOutboundRoute with the text "VASP:".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSDeliveryReport Charging Callback

This callback is executed when a delivery report is being routed by the MMSC.

NowSMS will ignore any HTTP response returned by the callback, however we recommend returning an "HTTP 200 OK" response for future compatibility reasons.

The following parameter variables may be set for the MMSDeliveryReport charging callback:

Type=MMSDeliveryReport

The transaction type is MMSDeliveryReport, indicating that an MMS Delivery Report has been generated.

From=SenderPhoneNumber

This parameter contains the phone number of the subscriber for which the delivery report has been generated (i.e., the original recipient of the message).

To=RecipientPhoneNumber

This parameter contains the phone number to which this delivery report is being sent (i.e., the original sender of the message).

VASP=MmscOutboundRoute

This parameter is present if the MMSC has determined that the delivery report must be routed via an external route for delivery. The value of this parameter refers to the account name as defined in the "MMSC Routing" list.

Note that some versions of NowSMS may preface the MmscOutboundRoute with the text "VASP:".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSReadReport PreAuth Callback

This callback is executed when a MMS subscriber, Value Added Service Provider (VASP) or MMSC interconnect partner is requesting to send a read report (a.k.a., message read receipt).

This is a “pre-authorisation” request, and does not mean that the read report will actually be accepted by NowSMS for delivery. If NowSMS cannot successfully connect to the accounting URL, or the URL returns a response other than a standard “HTTP 200 OK” response, the user request to send a message will be blocked. A “PreAuth” request to send a message will also be blocked if the HTTP response content includes the text “PreAuth=Deny”.

The following parameter variables may be set for the MMSReadReport pre-authorisation request:

PreAuth=Yes

The presence of this parameter indicates that this callback is a pre-authorisation request.

Type=MMSReadReport

The transaction type is MMSReadReport, indicating that a request is being made to send an MMS read report.

From=SenderPhoneNumber

This parameter contains the phone number of the subscriber for which the read report is being generated (i.e., the original recipient of the message).

To=RecipientPhoneNumber

This parameter contains the phone number to which this read report is being sent (i.e., the original sender of the message).

VASP=MmscOutboundRoute

This parameter is present if the MMSC has determined that the read report must be routed via an external route for delivery. The value of this parameter refers to the account name as defined in the "MMSC Routing" list.

Note that some versions of NowSMS may preface the MmscOutboundRoute with the text "VASP:".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSReadReport Charging Callback

This callback is executed when a read report is being routed by the MMSC.

NowSMS will ignore any HTTP response returned by the callback, however we recommend returning an "HTTP 200 OK" response for future compatibility reasons.

The following parameter variables may be set for the MMSReadReport charging callback:

Type=MMSReadReport

The transaction type is MMSReadReport, indicating that an MMS Read Report has been generated.

From=SenderPhoneNumber

This parameter contains the phone number of the subscriber for which the read report has been generated (i.e., the original recipient of the message).

To=RecipientPhoneNumber

This parameter contains the phone number to which this read report is being sent (i.e., the original sender of the message).

VASP=MmscOutboundRoute

This parameter is present if the MMSC has determined that the read report must be routed via an external route for delivery. The value of this parameter refers to the account name as defined in the "MMSC Routing" list.

Note that some versions of NowSMS may preface the MmscOutboundRoute with the text "VASP:".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSEMail PreAuth Callback

This callback is executed when an e-mail message has arrived via SMTP, specifying an MMS recipient.

This is a "pre-authorisation" request, and does not mean that the message will actually be accepted by NowSMS for delivery. If NowSMS cannot successfully connect to the accounting URL, or the URL returns a response other than a standard "HTTP 200 OK" response, the user request to send a message will be blocked. A "PreAuth" request to send a message will also be blocked if the HTTP response content includes the text "PreAuth=Deny".

The following parameter variables may be set for the MMSEMail pre-authorisation request:

PreAuth=Yes

The presence of this parameter indicates that this callback is a pre-authorisation request.

Type=MMSEMail

The transaction type is MMSEMail, indicating that an SMTP request is being made to deliver an MMS message to a subscriber.

From=EMailAddress

This parameter contains the e-mail address of the SMTP message sender.

To=RecipientPhoneNumber

This parameter contains a single recipient phone number.

MsgCount=1

This parameter is always 1 in current versions of NowSMS.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSEMail Charging Callback

This callback is executed when an SMTP message has been accepted for routing to an MMS recipient.

NowSMS will ignore any HTTP response returned by the callback, however we recommend returning an "HTTP 200 OK" response for future compatibility reasons.

The following parameter variables may be set for the MMSEMail Charging Callback:

Type=MMSEMail

The transaction type is MMSEMail, indicating that an SMTP message has been accepted for routing to an MMS recipient.

From=EMailAddress

This parameter contains the e-mail address of the SMTP message sender.

To=RecipientPhoneNumber

This parameter contains a single recipient phone number. If the original message was submitted to more than one recipient, a separate charging callback will occur for each recipient.

MessageID=assignedMessageID

This parameter records the MMS message ID assigned by MMSC. Note that if the message was sent to multiple recipients, each recipient instance shares the same message ID.

Size=####

This parameter specifies the size of the MMS message in bytes.

Note that MMS message size may differ based upon the encoding protocol (e.g., MM1, MM4, MM7). The actual size of the delivered MMS message may be different because of conversion between these protocols, and/or MMS header manipulation.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Query NowSMS Server Status

Starting with the NowSMS 2008 release, an XML-based status query interface is available which reports information similar to what is reported on the "Status" page of the NowSMS configuration dialog. The query results include information about SMSC connection status, the number of messages processed via the different connections, and the number of messages pending in the queues, among other information.

The XML-based status interface can be accessed on the web port of the SMS gateway using a URL of "/ADMIN/XMLSTATUS" (not case sensitive). To enable access to this interface, "Enable Web Account Administration" must be enabled on the "Web" tab of the NowSMS administrative interface. The Admin User Name and Password can either be sent in an HTTP "Authorization:" header, or using "&User=username&Password=password" in the URL request.

An example will help better illustrate the information that is available.

Connect to the NowSMS server with the following URL: http://127.0.0.1:8800/admin/xmlstatus?user=adminuser&password=adminpass Substitute in your appropriate IP address and port ... and the appropriate "adminuser" and "adminpass" values. The XML formatted information is quite readable, so a good way to get started is to make the request to your own NowSMS server via a web browser to see what information is returned.

Here's an example of the XML response with line breaks added for improved readability:

```
<NowSMSStatus>

<SMSCStatus>
<Name>SMPP - xps5:9000</Name>
<Status>OK</Status>
<MessagesToday>44640</MessagesToday>
<MessagesLast7Days>44640</MessagesLast7Days>
<MessagesLast30Days>44640</MessagesLast30Days>
</SMSCStatus>

<SMSCStatus>
<Name>SMPP - xps5:9001</Name>
<Status>OK</Status>
<MessagesToday>43369</MessagesToday>
<MessagesLast7Days>43369</MessagesLast7Days>
<MessagesLast30Days>43369</MessagesLast30Days>
</SMSCStatus>

<SMSCStatus>
<Name>SMPP - xps5#2:9000</Name>
<Status>OK</Status>
<MessagesToday>45308</MessagesToday>
<MessagesLast7Days>45308</MessagesLast7Days>
<MessagesLast30Days>45308</MessagesLast30Days>
</SMSCStatus>

<SMSCStatus>
<Name>SMPP - xps5#2:9001</Name>
<Status>OK</Status>
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
<MessagesToday>48856</MessagesToday>
<MessagesLast7Days>48856</MessagesLast7Days>
<MessagesLast30Days>48856</MessagesLast30Days>
</SMSCStatus>

<MMSCRouteStatus>
<Name>mm7test</Name>
<Status>OK</Status>
<MessagesToday>0</MessagesToday>
<MessagesLast7Days>5</MessagesLast7Days>
<MessagesLast30Days>5</MessagesLast30Days>
</MMSCRouteStatus>

<SMSOUTQ>92833</SMSOUTQ>
<SMSINQ>0</SMSINQ>
<SMSRCPTQ>0</SMSRCPTQ>
<MMSOUTQ>0</MMSOUTQ>

<SMSSubmitted>
<MessagesToday>275006</MessagesToday>
<MessagesLast7Days>275006</MessagesLast7Days>
<MessagesLast30Days>275006</MessagesLast30Days>
</SMSSubmitted>

<SMSSent>
<MessagesToday>182173</MessagesToday>
<MessagesLast7Days>182173</MessagesLast7Days>
<MessagesLast30Days>182173</MessagesLast30Days>
</SMSSent>

<SMSReceived>
<MessagesToday>73786</MessagesToday>
<MessagesLast7Days>73786</MessagesLast7Days>
<MessagesLast30Days>73786</MessagesLast30Days>
</SMSReceived>

<SMSFailed>
<MessagesToday>0</MessagesToday>
<MessagesLast7Days>0</MessagesLast7Days>
<MessagesLast30Days>0</MessagesLast30Days>
</SMSFailed>

<SMSRetried>
<MessagesToday>0</MessagesToday>
<MessagesLast7Days>0</MessagesLast7Days>
<MessagesLast30Days>0</MessagesLast30Days>
</SMSRetried>

<MMSProcessedUser>
<MessagesToday>0</MessagesToday>
<MessagesLast7Days>0</MessagesLast7Days>
<MessagesLast30Days>0</MessagesLast30Days>
</MMSProcessedUser>

<MMSProcessedVASP>
<MessagesToday>0</MessagesToday>
<MessagesLast7Days>0</MessagesLast7Days>
<MessagesLast30Days>0</MessagesLast30Days>
</MMSProcessedVASP>

<MMSSentMMSC>
<MessagesToday>0</MessagesToday>
<MessagesLast7Days>0</MessagesLast7Days>
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
<MessagesLast30Days>0</MessagesLast30Days>
</MMSSentMMSC>

<MMSSentVASP>
<MessagesToday>0</MessagesToday>
<MessagesLast7Days>0</MessagesLast7Days>
<MessagesLast30Days>0</MessagesLast30Days>
</MMSSentVASP>

<MMSRetrieved>
<MessagesToday>0</MessagesToday>
<MessagesLast7Days>0</MessagesLast7Days>
<MessagesLast30Days>0</MessagesLast30Days>
</MMSRetrieved>

<SMPPClientList>

<ActiveConnectionCount>2</ActiveConnectionCount>
<ActiveConnection>
<Name>micro</Name>
<ConnectionType>S</ConnectionType>
<IPAddress >192.168.1.114</IPAddress>
</ActiveConnection>

<ActiveConnection>
<Name>micro</Name>
<ConnectionType>R</ConnectionType>
<IPAddress >192.168.1.114</IPAddress>
</ActiveConnection>

</SMPPClientList>

</NowSMSStatus>
```

Most of the above information directly correlates to statistics that are displayed on the NowSMS configuration dialog.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Interfacing with NowSMS via PHP

Several example PHP scripts are available to simplify interfacing your application with NowSMS for sending and receiving messages. They are described on the following pages.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send SMS Text Message with PHP

The following example PHP script, sendsms.php, can be used to send an SMS text message via NowSMS with PHP. This script can also be downloaded from http://www.nowsms.com/download/sendsms-php.txt.

```
<?php

function SendSMS ($host, $port, $username, $password, $phoneNoRecip, $msgText) {

/* Parameters:
    $host - IP address or host name of the NowSMS server
    $port - "Port number for the web interface" of the NowSMS Server
    $username - "SMS Users" account on the NowSMS server
    $password - Password defined for the "SMS Users" account on the NowSMS Server
    $phoneNoRecip - One or more phone numbers (comma delimited) to receive the text
message
    $msgText - Text of the message
*/

    $fp = fsockopen($host, $port, $errno, $errstr);
    if (!$fp) {
        echo "errno: $errno \n";
        echo "errstr: $errstr\n";
        return $result;
    }

    fwrite($fp, "GET /?Phone=" . rawurlencode($phoneNoRecip) . "&Text=" .
rawurlencode($msgText) . " HTTP/1.0\n");
    if ($username != "") {
       $auth = $username . ":" . $password;
       $auth = base64_encode($auth);
       fwrite($fp, "Authorization: Basic " . $auth . "\n");
    }
    fwrite($fp, "\n");

    $res = "";

    while(!feof($fp)) {
        $res .= fread($fp,1);
    }
    fclose($fp);


    return $res;
}


/* This code provides an example of how you would call the SendSMS function from within
   a PHP script to send a message.  The response from the NowSMS server is echoed back
from the script.

$x   = SendSMS("127.0.0.1", 8800, "username", "password", "+44999999999", "Test Message");
echo $x;

*/
?>
```

The SendSMS function is the important part of the example. This is the function that needs to be included in your PHP script. You call this function, specifying the host name or IP address and port number of the NowSMS server, along with a username and password for an "SMS Users" account on the NowSMS server, plus the recipient phone number and text of the SMS message.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The SendSMS function uses these parameters to build a URL for connecting to the NowSMS server. This function could be easily modified to support sending other types of messages by modifying the URL that the function creates. For additional information on NowSMS URL parameters, see **Submitting SMS Messages - URL Parameters** on page 187.

As an additional example, here is how a web form might call a PHP script that uses sendsms.php to send an SMS message.

Below is a simple HTML web form that posts parameters to a PHP script named sendsmsscript.php:

```
<HTML>
<HEAD><TITLE>Send SMS</TITLE></HEAD>
<BODY>
<form method="post" action="sendsmsscript.php">
<table border="1">
<tr>
<td>Mobile Number:</td>
<td><input type="text" name="phone" size="40"></td>
</tr>
<tr>
<td valign="top">Text Message:</td>
<td><textarea name="text" cols="80" rows="10"></textarea>
</tr>
<tr>
<td colspan="2" align="center">
<input type="submit" value="Send">
</td>
</tr>
</table>
</form>
</BODY>
</HTML>
```

And here is an example sendsmsscript.php that processes the above form, validates the form parameters, and calls the SendSMS function from sendsms.php to submit the message:

```
if (isset($_REQUEST['phone'])) {
   if (isset($_REQUEST['text'])) {
      $x = SendSMS("127.0.0.1", 8800, "username", "password", $_REQUEST['phone'],
$_REQUEST['text']);
      echo $x;
   }
   else {
      echo "ERROR : Message not sent -- Text parameter is missing!\r\n";
   }
}
else {
   echo "ERROR : Message not sent -- Phone parameter is missing!\r\n";
}
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send OMA Client Provisioning with PHP

OMA Provisioing Content Messages are special SMS messages that contain information used to configure certain settings of a mobile phone, such as settings for the browser (APN, proxy, bookmarks), MMS client, IM client, or SyncML client.

These messages are sometimes referred to as OTA configuration messages, where OTA is an acronym for Over-The-Air.

These settings are specified in an XML format, and the encoded by NowSMS into a compact binary format for sending via SMS.

Some of the XML document formats supported by NowSMS include:

1.) OMA (Open Mobile Alliance) Provisioning Content (root XML element "wap-provisioningdoc")

2.) OMA (Open Mobile Alliance) DRM Rights Objects (root XML element "o-ex:rights")

3.) WAP Push Service Indication, Service Load and Cache Operation (root XML element "si", "sl", or "co")

4.) OMA (Open Mobile Alliance) E-Mail Notification (EMN) (root XML element "emn")

5.) Nokia/Ericsson Over The Air Settings (OTA) Specification (root XML element "CHARACTERISTIC-LIST")

6.) Nokia/Ericssson SyncML OTA or Wireless Village Settings (root XML element "SyncSettings" or "WVSettings")

The following example PHP script, sendota.php, can be used to send an any of these supported XML provisioning content types via NowSMS with PHP. This script can also be downloaded from http://www.nowsms.com/download/sendota-php.txt.

```
<?php

function SendOTA ($host, $port, $username, $password, $phoneNoRecip, $otafilename,
$otapin, $otapintype) {

/* Parameters:
    $host - IP address or host name of the NowSMS server
    $port - "Port number for the web interface" of the NowSMS Server
    $username - "SMS Users" account on the NowSMS server
    $password - Password defined for the "SMS Users" account on the NowSMS Server
    $phoneNoRecip - One or more phone numbers (comma delimited) to receive the text
message
    $otafilename - File name of a local file containing the XML document to be encoded
and sent as an OTA message
    $otapin - Optional PIN code for signing the OTA message
    $otapintype - Type of PIN code (USERPIN or NETWPIN)
*/

    $otadoc = file_get_contents ($otafilename);
    if (!$otadoc) {
        echo "ERROR:  Unable to open $otafilename\n";
        return null;
    }
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
    $fp = fsockopen($host, $port, $errno, $errstr);
    if (!$fp) {
        echo "errno: $errno \n";
        echo "errstr: $errstr\n";
        return null;
    }

    fwrite($fp, "POST /?Phone=" . rawurlencode($phoneNoRecip) . "&OTA=POST&OTAPIN=" .
rawurlencode($otapin) . "&OTAPINTYPE=" . rawurlencode($otapintype) . " HTTP/1.0\n");
    if ($username != "") {
       $auth = $username . ":" . $password;
       $auth = base64_encode($auth);
       fwrite($fp, "Authorization: Basic " . $auth . "\n");
    }
    fwrite($fp, "Content-length: " . strlen($otadoc) . "\n");
    fwrite($fp, "\n");
    fwrite($fp, $otadoc);

    $res = "";

    while(!feof($fp)) {
        $res .= fread($fp,1);
    }
    fclose($fp);


    return $res;
}

/* This code provides an example of how you would call the SendOTA function from within
   a PHP script to send a message.  The response from the NowSMS server is echoed back
from the script.
*/
$x   = SendOTA("127.0.0.1", 8800, "username", "password", "+4499999999999",
"f:/temp/bookmark.xml", "0000", "USERPIN");
echo $x;


?>
```

This PHP script contains a function called SendOTA.

The function definition for SendOTA contains the following parameters:

function SendOTA ($host, $port, $username, $password, $phoneNoRecip, $otafilename, $otapin, $otapintype)

$host - IP address or host name of the NowSMS server
$port - "Port number for the web interface" of the NowSMS Server
$username - "SMS Users" account on the NowSMS server
$password - Password defined for the "SMS Users" account on the NowSMS Server
$phoneNoRecip - One or more phone numbers (comma delimited) to receive the text message
$otafilename - File name of a local file containing the XML document to be encoded and sent as an OTA message
$otapin - Optional PIN code for signing the OTA message
$otapintype - Type of PIN code (USERPIN or NETWPIN)

Assuming that you have created an XML document named f:/temp/bookmark.xml, the following function call could be used to submit this document for sending as an OTA message via NowSMS:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
SendOTA("127.0.0.1", 8800, "username", "password", "+4499999999999",
"f:/temp/bookmark.xml", "0000", "USERPIN");
```

PIN codes are always the most confusing part of sending an OTA message.

An OTA PIN can be associated with many of the XML settings types to provide a layer of authentication to the message. Many devices will allow you to send XML settings without a PIN, but some will require a PIN to be present before the settings will be accepted at all.

There are three different types of PINs, depending on the "OTA PIN Type" ($otapintype) setting.

The simplest "OTA PIN Type" is "USERPIN" (User PIN). This setting indicates that a short PIN code (often 4 digits) is supplied as the "OTA PIN". When the user receives the OTA settings message, they will need to supply this PIN code in order to be able to open the message and apply the settings. If a "PINValue" is specified, but a "PINType" is not, then USERPIN will be assumed.

"NETWPIN" (Network PIN) indicates the PIN is a network PIN code. In the GSM environment, this is the IMSI number associated with the SIM card in the device. (Hint, if you want to experiment with determining the PIN card associated with a SIM, you can put the SIM into a GSM modem and the AT+CIMI command to return the IMSI. However, not all GSM modems support the AT+CIMI command.) When the device receives the settings, if the NETWPIN does not match the IMSI, the settings will be discarded.

An additional type of PIN, known as "USERNETWPIN" also exists, which indicates a combination of the USERPIN and NETWPIN types. To use this OTA PIN type, specify the OTA PIN as the IMSI number associated with the SIM card in the device, followed by a ":" character, followed by a USERPIN (e.g., 1234567889012345:1234). When the device receives the settings, the user will be prompted for a PIN. This user supplied PIN, and the SIM card IMSI, must match in order for the settings to be accepted.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send MMS Message with PHP

For details, please see **Send MMS Message with PHP** on page 166.

Receive MMS Message with PHP

For details, please see **Receiving MMS Messages with a PHP Script: HTTP File Upload Post** on page 391.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Interfacing with NowSMS via Java

Several example Java classes are available to simplify interfacing your application with NowSMS for sending and receiving messages. They are described on the following pages.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send SMS Text Message with Java

The following example Java class, sendsms, can be used to send an SMS text message via NowSMS from Java. This script can also be downloaded from http://www.nowsms.com/download/sendsms.java.txt .

```
import java.net.*;
import java.io.*;

public class sendsms {


    public static String server;
    public static String user;
    public static String password;
    public static String phonenumber;
    public static String text;
    public static String data;
    public static String udh;
    public static String pid;
    public static String dcs;
    public static String sender;
    public static String validity;
    public static String servicetype;
    public static String smscroute;
    public static String receiptrequested;
    public static String sourceport;
    public static String destport;
    public static String delayuntil;
    public static String voicemail;
    public static String wapurl;
    public static String wapsl;

    public static String url_str;

    public static void init () {
        server = null;
        user = null;
        password = null;
        phonenumber = null;
        text = null;
        data = null;
        udh = null;
        pid = null;
        dcs = null;
        sender = null;
        validity = null;
        servicetype = null;
        smscroute = null;
        receiptrequested = null;
        sourceport = null;
        destport = null;
        delayuntil = null;
        voicemail = null;
        wapurl = null;
        wapsl = null;
    }

    public static void setvar (String argname, String argvalue) {

       if (argname != null) {
          if (argvalue != null) {
             url_str = url_str + "&" + argname + "=";
             try {
                String encoded = URLEncoder.encode (argvalue, "UTF-8");
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
               url_str = url_str + encoded;
            }
            catch (UnsupportedEncodingException e) {
               url_str = url_str + argvalue;
            }
         }
      }

   }

   public static String send () {


      String returnstring;

      returnstring = null;

      if (server == null) {
        System.out.println("sendsms.server value not set");
         return returnstring;
      }

      url_str = server + "?";
      setvar("user", user);
      setvar("password", password);
      setvar("phonenumber", phonenumber);
      setvar("text", text);
      setvar("data", data);
      setvar("udh", udh);
      setvar("pid", pid);
      setvar("dcs", dcs);
      setvar("sender", sender);
      setvar("validity", validity);
      setvar("servicetype", servicetype);
      setvar("smscroute", smscroute);
      setvar("receiptrequested", receiptrequested);
      setvar("sourceport", sourceport);
      setvar("destport", destport);
      setvar("delayuntil", delayuntil);
      setvar("voicemail", voicemail);
      setvar("wapurl", wapurl);
      setvar("wapsl", wapsl);

      try {
         URL url2=new URL(url_str);

         HttpURLConnection connection = (HttpURLConnection) url2.openConnection();
         connection.setDoOutput(false);
         connection.setDoInput(true);

         String res=connection.getResponseMessage();

         System.out.println("Response Code  ->"+res);

         int code = connection.getResponseCode () ;

         if ( code == HttpURLConnection.HTTP_OK ) {
            //Get response data.
            BufferedReader in = new BufferedReader(new
InputStreamReader(connection.getInputStream()));

            String str;

            while( null != ((str = in.readLine()))) {
               if (str.startsWith("MessageID=")) {
                  returnstring = returnstring + str + "\r\n";
                  System.out.println(str);
               }
            }
            connection.disconnect() ;
         }
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
        }
        catch(IOException e) {
           System.out.println("unable to create new url"+e.getMessage());
        }
        return returnstring;
    }
}
```

The sendsms class defined in this example can be used to send simple SMS text messages, as well as many types of binary SMS messages, including WAP Push.

The class supports many of the URL parameters that are defined for NowSMS, and could easily be adapted to support additional parameters.

NowSMS URL parameters are supported as methods for the sendsms class, with method names matching the URL parameter names, except that all methods are in lower case.

In addition the URL parameter methods, the following additional methods are defined:

```
sendsms.init();
```

The init method initialise the SMS message object.

```
sendsms.server = "http://localhost:8800/";
```

The server method sets the URL address for the NowSMS server.

```
sendsms.send();
```

The send method submits the SMS message to NowSMS. (The send method returns a list of message ids assigned for the submitted message, with one message id per line, in the following format: MessageID=xxxxxxxxxxxxxx.req, Recipient=xxxxxxxxxxx)

Supported methods for setting URL parameters:

```
sendsms.user = "username";
sendsms.password = "password";
```

These methods specify the authorisation credentials of the "SMS Users" account that is submitting the message.

```
sendsms.phonenumber = "recipient";
```

This method specifies the recipient phone number to which this message should be addressed. This can be a comma delimited list of recipient phone numbers or a distribution list name.

```
sendsms.text = "text of message";
```

This method specifies the text of the message. (Or for WAP Push messages, the text associated with the WAP Push URL.)

```
sendsms.data = "hexstring";
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

This method specifies a hex string of binary data for sending a binary SMS message.

```
sendsms.udh = "hexstring";
```

This method specifies a hex string of binary data for the User Data Header (UDH) of the SMS message.

```
sendsms.pid = "PID Value";
```

This method specifies a hex value representing the SMS Protocol ID (PID) of this SMS message.

```
sendsms.dcs = "DCS Value";
```

This method specifies a hex value representing the value of the SMS Data Coding Scheme (DCS) for this message.

```
sendsms.sender = "Sender";
```

This method specifies the sender (source) address to be specified for this message. (Note: It is not possible to override the sender address when sending messages via a GSM modem.)

```
sendsms.smscroute = "route name";
```

This method specifies an outbound SMSC route to be used for the message. (For more information on SMS message routing, see SMS Message Routing Logic on page 54.)

```
sendsms.receiptrequested = "Yes"';
```

This method can be used to request a delivery receipt.

```
sendsms.sourceport = "3333";
sendsms.destport = "3333";
```

These methods can be used to specify source and destination ports for routing the SMS message to a specific application on the recipient mobile phone.

```
sendsms.delayuntil = "YYYYMMDDHHMM";
```

This method allows messages to be submitted to NowSMS and queued for later processing. The value of this parameter should be of the format "YYYYMMDDHHMM", indicating the date and time until which the message should be delayed, where YYYY is the year, MM is the month, DD is the day, HH is the hour (in 24 hour format), and MM is the minutes.

```
sendsms.wapurl = "http://x/path";
```

Specifies that the a WAP Push message should be sent, and this is the URL to be sent in the WAP Push message.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Example - Sending a simple text message:

```
sendsms.init();
sendsms.server = "http://127.0.0.1:8800/";
sendsms.user = "test";
sendsms.password = "test";
sendsms.phonenumber = "+9999999999";
sendsms.text = "This is a test message";
sendsms.send();
```

Example - Sending a text message to a specific application port for a Java applet running on the phone:

```
sendsms.init();
sendsms.server = "http://127.0.0.1:8800/";
sendsms.user = "test";
sendsms.password = "test";
sendsms.phonenumber = "+9999999999";
sendsms.text = "This is a test message";
sendsms.destport = "3333";
sendsms.send();
```

Example - Sending a WAP Push Message:

```
sendsms.init();
sendsms.server = "http://127.0.0.1:8800/";
sendsms.user = "test";
sendsms.password = "test";
sendsms.phonenumber = "+9999999999";
sendsms.text = "This is a test message";
sendsms.wapurl = "http://www.nowsms.com/";
sendsms.send();.
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send MMS Message with Java

For more information, please see Send MMS Message with Java on page 172.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Interfacing with NowSMS via Command Line Interface

One of the simplest ways to send SMS or MMS messages with NowSMS is via a command line interface. Several command line scripts for Windows are provided to allow messages to be submitted to NowSMS from either the local PC running NowSMS, or any other PC with a network connection.

Send SMS Text Message from the Command Line

For details, please see Sending SMS from the Command Line on page 376.

Send MMS Message from the Command Line

For details, please see Send MMS Message from Command Line on page 176.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send WAP Push and Binary SMS from the Command Line

The following Windows Jscript example can be used to enable sending of WAP Push and other binary SMS messages from a command line interface.

```
/*
This is a command line script for sending an SMS message via NowSMS.

Below, you must substitute in the address of your NowSMS server, plus a valid
username and password for
an account defined in the "SMS Users" list of that server.

Note: This script uses the encodeURIComponent method introduced in Internet
Explorer 5.5. If you are running
Windows 2000 or an earlier version of Windows, you must have Internet Explorer
5.5 or later installed for this
script to work.
*/

var NowSMSServerAddress = "http://127.0.0.1:8800";
var NowSMSUserName = "test";
var NowSMSPassword = "test";

function HTTPGET(strURL)
{
   var strResult;

   try
   {
      // Create the WinHTTPRequest ActiveX Object.
      var WinHttpReq = new ActiveXObject("Msxml2.XMLHTTP" /* or
"WinHttp.WinHttpRequest.5"*/);

      // Create an HTTP request.
      var temp = WinHttpReq.Open("GET", strURL, false);

      // Send the HTTP request.
      WinHttpReq.Send();

      // Retrieve the response text.
      strResult = WinHttpReq.ResponseText;
   }
   catch (objError)
   {
      strResult = objError + "\n"
      strResult += "WinHTTP returned error: " + (objError.number &
0xFFFF).toString() + "\n\n";
      strResult += objError.description;
   }

   // Return the response text.
   return strResult;
}

var strRequest;

if (WScript.Arguments.Count() < 2) {
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
   WScript.Echo ("Usage: " + WScript.ScriptName + "
PhoneNumber1[,PhoneNumber2,...] NowSMSURLParameter1=Value1
[NowSMSURLParameter2=Value2] [NowSMSURLParameterN=ValueN]\r\n");
   WScript.Quit();
}

strRequest = NowSMSServerAddress + "/?PhoneNumber=" +
encodeURIComponent(WScript.Arguments(0)) + "&User=" +
encodeURIComponent(NowSMSUserName) + "&password=" +
encodeURIComponent(NowSMSPassword) ;

for (i=1; i<WScript.Arguments.Count(); i++)
{
   var strTemp = WScript.Arguments(i);

   while (strTemp.indexOf("^") >= 0) {
      var temp = strTemp.indexOf("^");
      if (temp >= 0) {
         strTemp = strTemp.substr(0, temp) + strTemp.substr(temp+1);
      }
   }

   var equPos = strTemp.indexOf("=");
   if (equPos >= 0) {
      strRequest += "&";
      strRequest += strTemp.substr(0,equPos+1);
      strRequest += encodeURIComponent(strTemp.substr(equPos+1));
   }
   else {
      strRequest += "%20";
      strRequest += encodeURIComponent(strTemp);
   }

}


/* WScript.Echo(strRequest);  */

WScript.Echo(HTTPGET(strRequest));
```

This script can be downloaded from http://www.nowsms.com/download/sms2.js.txt.

Assuming that the script file is saved as a file named sms2.js, you would issue the following command:

```
cscript sms2.js PhoneNumber1[,PhoneNumber2,...] NowSMSURLParameter1=Value1
[NowSMSURLParameter2=Value2] [NowSMSURLParameterN=ValueN]
```

(cscript.exe is the Windows Script Host, which is a component of Windows which should be located in the \Windows\System32 directory.)

NowSMSURLParameter1=Value1 can be any of the URL parameters documented for NowSMS in **Submitting SMS Messages - URL Parameters** on page 187.

However, one difference from the URL parameter interface is that this script will perform any necessary URL escaping to simplify the user interface to the script.

Multiple URL parameters can be specified via this command line interface.

For example, to send a WAP push message, the following command could be used:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
cscript sms2.js +44777777777 WAPURL=http://www.nowsms.com "Text=This is a test
push"
```

In this example, because the "Text=" parameter includes spaces, we put quotes around the paramter and value, so that it is recognised as a single parameter value combination.

A simpler example would be sending a message to turn on the voice message waiting indicator (MWI):

```
cscript sms2.js +44777777777 VoiceMail=On
```

Or, if you have a pre-constructed binary message, you can directly specify the UDH and Data parameters, such as in the following example:

```
cscript sms2.js +44777777777 Binary=1 UDH=080D0200040B020007 Data=20 DCS=8
```

A more complex example would be sending an EMS message. NowSMS defines simple mark-up tags for inserting attributes into an EMS text message. For example, "This is a test!" encodes the text with "test" highlighted as bold on supported EMS compatible phones.

Unfortunately, the "<" and ">" characters are used for input/output redirection from the Windows command line. Therefore to include those characters in a command line string, it is necessary to preface the character with the escape character "^".

To send this EMS message, the following command line would be used:

```
cscript sms2.js +44777777777 "EMSText=This is a ^<b^>test^</b^>!"
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Send OMA Client Provisioning from the Command Line

OMA Provisioing Content Messages are special SMS messages that contain information used to configure certain settings of a mobile phone, such as settings for the browser (APN, proxy, bookmarks), MMS client, IM client, or SyncML client.

These messages are sometimes referred to as OTA configuration messages, where OTA is an acronym for Over-The-Air.

These settings are specified in an XML format, and the encoded by NowSMS into a compact binary format for sending via SMS.

Some of the XML document formats supported by NowSMS include:

1.) OMA (Open Mobile Alliance) Provisioning Content (root XML element "wap-provisioningdoc")

2.) OMA (Open Mobile Alliance) DRM Rights Objects (root XML element "o-ex:rights")

3.) WAP Push Service Indication, Service Load and Cache Operation (root XML element "si", "sl", or "co")

4.) OMA (Open Mobile Alliance) E-Mail Notification (EMN) (root XML element "emn")

5.) Nokia/Ericsson Over The Air Settings (OTA) Specification (root XML element "CHARACTERISTIC-LIST")

6.) Nokia/Ericssson SyncML OTA or Wireless Village Settings (root XML element "SyncSettings" or "WVSettings")

The following example Jscript, sendota.js, can be used to send an any of these supported XML provisioning content types via NowSMS from a command line interface. This script can also be downloaded from http://www.nowsms.com/download/sendota.js.txt.

```
/*
This is a command line script for sending an OMA Client Provisioning / XML Settings
Document via NowSMS.

Below, you must substitute in the address of your NowSMS server, plus a valid username
and password for
an account defined in the "SMS Users" list of that server.

Note: This script uses the encodeURIComponent method introduced in Internet Explorer 5.5.
If you are running
Windows 2000 or an earlier version of Windows, you must have Internet Explorer 5.5 or
later installed for this
script to work.
*/

var NowSMSServerAddress = "http://127.0.0.1:8800";
var NowSMSUserName = "test";
var NowSMSPassword = "test";

function HTTPPOST(strURL,xmlSettings)
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
{
   var strResult;

   try
   {
      // Create the WinHTTPRequest ActiveX Object.
      var WinHttpReq = new ActiveXObject("Msxml2.XMLHTTP" /* or
"WinHttp.WinHttpRequest.5"*/);

      // Create an HTTP request.
      var temp = WinHttpReq.Open("POST", strURL, false);

      // Send the HTTP request.
      WinHttpReq.Send(xmlSettings);

      // Retrieve the response text.
      strResult = WinHttpReq.ResponseText;
   }
   catch (objError)
   {
      strResult = objError + "\n"
      strResult += "WinHTTP returned error: " + (objError.number & 0xFFFF).toString() +
"\n\n";
      strResult += objError.description;
   }

   // Return the response text.
   return strResult;
}

var objFSO, objTextFile;
var ForReading = 1;
var strRequest;
var xmlSettings;

if (WScript.Arguments.Count() < 2) {
   WScript.Echo ("Usage: " + WScript.ScriptName + " PhoneNumber1[,PhoneNumber2,...]
OTAFilename [PINValue] [USERPIN | NETWPIN]\r\n");
   WScript.Quit();
}

objFSO = new ActiveXObject("Scripting.FileSystemObject");
try {
   objTextFile = objFSO.OpenTextFile(WScript.Arguments(1), ForReading);
}
catch (objError) {
   WScript.Echo ("Cannot open file " + WScript.Arguments(1) + "\r\n");
   WScript.Quit();
}

xmlSettings = objTextFile.ReadAll();

objTextFile.Close();

strRequest = NowSMSServerAddress + "/?PhoneNumber=" +
encodeURIComponent(WScript.Arguments(0)) + "&User=" + encodeURIComponent(NowSMSUserName)
+ "&password=" + encodeURIComponent(NowSMSPassword);

if (WScript.Arguments.Count() > 2) {
   strRequest += "&OTAPIN=" + encodeURIComponent(WScript.Arguments(2));
}
if (WScript.Arguments.Count() > 3) {
   strRequest += "&OTAPINTYPE=" + encodeURIComponent(WScript.Arguments(3));
}


strRequest += "&ota=post";

WScript.Echo(HTTPPOST(strRequest,xmlSettings));
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Assuming that the script file is saved as a file named smsota.js, you would issue the following command:

```
cscript sms.js PhoneNumber1[,PhoneNumber2,...] OTAFilename.xml [PINValue]
[USERPIN NETWPIN USERNETWPIN]
```

(cscript.exe is the Windows Script Host, which is a component of Windows which should be located in the \Windows\System32 directory.)

Examples:

```
cscript smsota.js +44777777777 settings.xml
cscript smsota.js +44777777777,+44777777778 settings.xml
cscript smsota.js +44777777777 settings.xml 1234
cscript smsota.js +44777777777 settings.xml 1234 USERPIN
```

PIN codes are always the most confusing part of sending an OTA message.

An OTA PIN can be associated with many of the XML settings types to provide a layer of authentication to the message. Many devices will allow you to send XML settings without a PIN, but some will require a PIN to be present before the settings will be accepted at all.

There are three different types of PINs, depending on the "OTA PIN Type" setting.

The simplest "OTA PIN Type" is "USERPIN" (User PIN). This setting indicates that a short PIN code (often 4 digits) is supplied as the "OTA PIN". When the user receives the OTA settings message, they will need to supply this PIN code in order to be able to open the message and apply the settings. If a "PINValue" is specified, but a "PINType" is not, then USERPIN will be assumed.

"NETWPIN" (Network PIN) indicates the PIN is a network PIN code. In the GSM environment, this is the IMSI number associated with the SIM card in the device. (Hint, if you want to experiment with determining the PIN card associated with a SIM, you can put the SIM into a GSM modem and the AT+CIMI command to return the IMSI. However, not all GSM modems support the AT+CIMI command.) When the device receives the settings, if the NETWPIN does not match the IMSI, the settings will be discarded.

An additional type of PIN, known as "USERNETWPIN" also exists, which indicates a combination of the USERPIN and NETWPIN types. To use this OTA PIN type, specify the OTA PIN as the IMSI number associated with the SIM card in the device, followed by a ":" character, followed by a USERPIN (e.g., 1234567889012345:1234). When the device receives the settings, the user will be prompted for a PIN. This user supplied PIN, and the SIM card IMSI, must match in order for the settings to be accepted.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Digital Rights Management

NowSMS provides full support for OMA Digital Rights Management (DRM) v1.0, with support for forward lock, combined delivery and separate delivery.

"Forward Lock" is the most basic level of DRM. When "Forward Lock" is enabled, this indicates that the receiving device should not allow any non-text objects in the message to be forwarded off of the device. The device may allow the user to extract pictures, videos or sounds from the message and save them on the phone. However, any such objects remain forward locked, such that they cannot be forwarded to another user or transferred to another device.

More advanced DRM restrictions can be applied to limit the number of times that the user can access an object, or start and end dates can be specified to limit how long the user can access an object. The restrictions can only be applied when using either "Combined Delivery" or "Separate Delivery".

When using either "Combined Delivery" or "Separate Delivery", a rights object is associated with the DRM protected object. This rights object defines permissions and constraints for how the DRM protected object may be used by the receiver.

"Combined Delivery" means that both the protected object and the rights object are delivered simultaneously. The protected object and its rights object are packaged as a multipart/related MIME object, which is usually retrieved by the receiving client via HTTP download or as part of the content of an MMS message.

"Separate Delivery" means that the protected object and the rights object are delivered separately. The protected object is retrieved by the receiving client via HTTP download or as part of the content of an MMS message. However, the protected object is locked and cannot be used until a rights object is separately delivered. In this case, the rights object is usually sent via WAP Push over SMS (sometimes premium rate SMS).

When using "Forward Lock" or "Combined Delivery", the DRM protected object is not actually encrypted. The protocol was designed in this lightweight fashion in order to allow DRM to be implemented on low end devices with limited capabilities.

Only when a DRM protected object is sent using "Separate Delivery" can the protected object actually be delivered to the device in an encrypted format.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

DRM with the NowSMS Web Interface

The web interface of NowSMS offers built-in support using "Forward Lock" or "Combined Delivery" when sending MMS or Multimedia WAP Push content. When sending an either of these message types, it is possible to specify Digital Rights Management (DRM) restrictions over the content of the message.

The most basic level of DRM is forward locking. When "**Forward Lock**" is set to "Yes", this indicates that the receiving device should not allow any non-text objects in the message to be forwarded off of the device. The device may allow the user to extract pictures, videos or sounds from the message and save them on the phone. However, any such objects remain forward locked, such that they cannot be forwarded to another user or transferred to another device.

More advanced DRM restrictions can be applied to limit the number of times that the user can access an object, or start and end dates can be specified to limit how long the user can access an object.

These advanced DRM restrictions can be applied by setting "**Restrict Content w/ DRM**" to "Yes". When this setting is enabled, forward lock is also implied, and the value of the "Forward Lock" setting is ignored.

"**DRM Permissions**" specify what types of access are allowed against the object. For example, an audio or video object requires "**Play**" permission before the user can access it. An image requires "**Display**" permission before the user can access it, and it requires "**Print**" permission if the user is to be allowed to print the image to a printer , perhaps over Bluetooth. An application requires "**Execute**" permission before the user can make use of the application. In all cases, the forward locking is assumed, so that the user is not allowed to forward or transfer the object from the device.

If you are sending multiple types of objects in the MMS message, check all permissions that are required for the different types of objects that you are sending.

"**DRM Constraints**" specify constraints with regard to how long the object should remain accessible to the user. It is possible to specify one or more of these constraints.

"**# of Accesses (count)**" specifies the the user can only access the object this number of times before access is no longer allowed.

"**Start Date (yyyy-mm-dd)**" specifies that the user will not be allowed to access the object until on or after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2006-02-24.)

"**End Date (yyyy-mm-dd)**" specifies that the user will not be allowed to access the object after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2006-02-24.)

"**# of Days (interval)**" specifies that the user will be allowed to access the object for this number of days after initial receipt of the object. The user can either enter a number of days here, or they can enter any valid value defined for the "<interval>" element in the

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

OMA DRM Rights Expression Language specification. For example, P2Y10M15DT10H30M20S represents a duration of 2 years, 10 months, 15 days, 10 hours, 30 minutes and 20 seconds.

DRM when interfacing with NowSMS Programmatically

NowSMS also provides support for sending DRM protected objects when interfacing to NowSMS programmatically.

When submitting an MMS message or Multimedia WAP Push message, it is possible to set parameters to indicate that the message content should be protected with DRM. This support is only available when messages are submitted via the **Now SMS/MMS Proprietary URL Submission** method.

Of course, NowSMS also supports DRM protected content that is submitted via the standardised MMS protocols (i.e., MM7, MM4, MM1). However, when these standard protocols are used, it is required that the content already be wrapped with DRM protection before it is submitted to NowSMS. When the Now SMS/MMS Proprietary URL Submission method is used, NowSMS can apply the DRM protection automatically.

The following section will detail extensions to the Now SMS/MMS Proprietary URL Submission method which allow for NowSMS to apply a DRM wrapper to the message content.

This will be followed by a section that describes the DRMCOMP command line utility which is a command line utility for applying DRM protection to an object. The DRMCOMP utility can be used to create protected objects that can be sent using DRM "Separate Delivery", in addition to support for the "Combined Delivery and "Forward Lock" formats.

DRM Forward Lock and Combined Delivery

When submitting an MMS message or Multimedia WAP Push message to NowSMS using the **Now SMS/MMS Proprietary URL Submission** method *(page 178)*, it is possible to specify additional parameters that are used to specify DRM protection.

These parameters can be specified either as part of the URL request (e.g., &parameter=value), or using the HTTP POST multipart/form-data encoding.

MMSForwardLock	MMS Message, Multimedia WAP Push	Forward locking is the most basic level of DRM (Digital Rights Management). When "Forward Lock" is set to "Yes", this indicates that the receiving device should not allow any non-text objects in the message to be forwarded off of the device. The device may allow the user to extract pictures, videos or sounds from the

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

		message and save them on the phone. However, any such objects remain forward locked, such that they cannot be forwarded to another user or transferred to another device.
DRMRestrict	MMS Message, Multimedia WAP Push	Beyond forward locking, More advanced DRM (Digital Rights Management) restrictions can be applied to limit the number of times that the user can access an object, or start and end dates can be specified to limit how long the user can access an object. These advanced DRM restrictions can be applied by setting "DRMRestrict" to "Yes". When this setting is enabled, forward lock is also implied, and the value of the "MMSForwardLock" setting is ignored.
DRMRestrictTextXML	MMS Message, Multimedia WAP Push	"Yes" specifies that the rights object should be encoded in text XML format. "No" specfies that the rights object should be encoded in binary XML format. The default is "No".
		"DRM Permissions" specify what types of access are allowed against the objects in a message that is protected with DRM. For example, an audio or video object requires "Play" permission before the user can access it. An image requires "Display" permission before the user can access it, and it requires "Print" permission if the user is to be allowed to print the image to a printer , perhaps over Bluetooth. An application requires "Execute" permission before the user can make use of the application. In all cases, the forward locking is assumed, so that the user is not allowed to forward or transfer the object from the device. If you are sending multiple types of objects in the MMS message, check all permissions that are required for the different types of objects that are being sent.
DRMPermissionPlay	MMS Message,	Set to "Yes" to enable DRM "Play"

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

	Multimedia WAP Push	Permission.
DRMPermissionDisplay	MMS Message, Multimedia WAP Push	Set to "Yes" to enable DRM "Display" Permission.
DRMPermissionExecute	MMS Message, Multimedia WAP Push	Set to "Yes" to enable DRM "Execute" Permission.
DRMPermissionPrint	MMS Message, Multimedia WAP Push	Set to "Yes" to enable DRM "Print" Permission.
		"DRM Constraints" specify constraints with regard to how long a DRM protected object object should remain accessible to the user. It is possible to specify one or more of these constraints that follow.
DRMConstraintCount	MMS Message, Multimedia WAP Push	"# of Accesses (count)" specifies the the user can only access the DRM protected object this number of times before access is no longer allowed.
DRMConstraintStart	MMS Message, Multimedia WAP Push	"Start Date (yyyy-mm-dd)" specifies that the user will not be allowed to access the DRM protected object until on or after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2006-02-24.)
DRMConstraintEnd	MMS Message, Multimedia WAP Push	"End Date (yyyy-mm-dd)" specifies that the user will not be allowed to access the DRM protected object after the specified date. (Note that you must specify the date in yyyy-mm-dd format, e.g., 2006-02-24.)
DRMConstraintInterval	MMS Message, Multimedia WAP Push	"# of Days (interval)" specifies that the user will be allowed to access the DRM protected object for this number of days after initial receipt of the object. The user can either enter a number of days here, or they can enter any valid value defined for the "<interval>" element in the OMA DRM Rights Expression Language specification. For example, P2Y10M15DT10H30M20S represents a duration of 2 years, 10 months, 15 days, 10 hours, 30 minutes and 20 seconds.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

DRMCOMP Utility & DRM Separate Delivery

The DRMCOMP command line utility can be used to add a wrapper of DRM protection to a content object. DRM wrappers are added to individual pieces of content, such as an image or video, and not to an entire MMS message.

The DRMCOMP command line utility can be used to add DRM wrappers for forward lock, combined delivery or separate delivery.

DRMCOMP - Forward Lock

```
Usage: (Forward lock)
   DRMCOMP objectFilename -forwardlock [-contentType=type/subtype]
                                       [-out=outputFilename]
```

`objectFilename` - This is the name of the file that is to have the DRM wrapper applied.

`-forwardlock` - This parameter indicates that a forward lock DRM wrapper should be added to the message.

`-contentType=type/subtype` - This optional parameter defines the MIME content type associated with the object (e.g., image/jpeg, audio/mp3). If this parameter is not specified, DRMCOMP will attempt to identify the content type automatically based upon the file extension. To determine the file extension to MIME type mapping, DRMCOMP will consult the NowSMS MMSCTYPE.INI file, if present, and will then search the MIME file extension to content type database in the Windows registry.

`-out=outputFilename` - This optional parameter specifies the output file name to contain the content with the DRM wrapper. If no filename is specified, DRMCOMP will default to the input objectFilename parameter, changing the file extension to ".dm".

Forward-lock DRM objects have a MIME type of "application/vnd.oma.drm.message". Note, however, that if you store a forward lock file on a conventional web server, some mobile phones may not properly understand the content. The reason for this is because the "application/vnd.oma.drm.message" format uses MIME multipart encoding. The Content-Type header returned from the web server should indicate, Content-Type: application/vnd.oma.drm.message; boundary="drm-boundary". It is not possible to configure most web servers to return a boundary parameter. While the OMA specifications suggest that a device should be able to understand a DRM object even if the boundary parameter is not present in the Content-Type header, this could present potential problems.

For convenience, DRMCOMP uses a fixed boundary parameter of "--generic-drm-boundary". Therefore, the Content-Type header for any forward-lock DRM objects created by DRMCOMP should be:

```
Content-Type: application/vnd.oma.drm.message; boundary="--generic-
drm-boundary"
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

DRMCOMP - Combined Delivery

```
Usage: (Combined delivery)
   DRMCOMP objectFilename -combineddelivery [-permission=play]
                                            [-permission=display]
                                            [-permission=execute]
                                            [-permission=print]
                                            [-limitCount=###]
                                            [-limitStartDate=yyyymmdd]
                                            [-limitEndDate=yyyymmdd]
                                            [-limitDays=###]
                                            [-contentType=type/subtype]
                                            [-contentID=cid]
                                            [-textXML]
                                            [-out=outputFilename]
```

`objectFilename` - This is the name of the file that is to have the DRM wrapper applied.

`-combinedDelivery` - This parameter indicates that a combined delivery DRM wrapper should be added to the message.

"DRM Permissions" specify what types of access are allowed against the objects in a message that is protected with DRM. The "permission=" parameters specify these permissions.

For example, an audio or video object requires "Play" permission before the user can access it. An image requires "Display" permission before the user can access it, and it requires "Print" permission if the user is to be allowed to print the image to a printer , perhaps over Bluetooth. An application requires "Execute" permission before the user can make use of the application. In all cases, the forward locking is assumed, so that the user is not allowed to forward or transfer the object from the device.

`-permission=play` - Enables "Play" permission for the protected object.

`-permission=display` - Enables "Display" permission for the protected object.

`-permission=execute` - Enables "Execute" permission for the protected object.

`-permission=print` - Enables "Print" permission for the protected object.

"DRM Constraints" specify constraints with regard to how long a DRM protected object object should remain accessible to the user.

It is possible to specify one or more of these "limit" constraints that follow.

`-limitCount=###` - Specifies the the user can only access the DRM protected object this number of times before access is no longer allowed

`-limitStartDate=yyyymmdd` - Specifies that the user will not be allowed to access the DRM protected object until on or after the specified date.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

-limitEndDate=yyyymmdd - Specifies that the user will not be allowed to access the DRM protected object after the specified date

-limitDays=### - Specifies that the user will be allowed to access the DRM protected object for this number of days after initial receipt of the object. The user can either enter a number of days here, or they can enter any valid value defined for the "<interval>" element in the OMA DRM Rights Expression Language specification. For example, P2Y10M15DT10H30M20S represents a duration of 2 years, 10 months, 15 days, 10 hours, 30 minutes and 20 seconds.

-contentType=type/subtype - This optional parameter defines the MIME content type associated with the object (e.g., image/jpeg, audio/mp3). If this parameter is not specified, DRMCOMP will attempt to identify the content type automatically based upon the file extension. To determine the file extension to MIME type mapping, DRMCOMP will consult the NowSMS MMSCTYPE.INI file, if present, and will then search the MIME file extension to content type database in the Windows registry.

-contentID=cid - This optional parameter defines a "Content-ID:" header to be included to identify the object. If this parameter is not present, DRMCOMP will generate one automatically. Should the object expire due to DRM constraints, it is possible to send a separate delivery rights object to re-enable this object, but the separate delivery rights object needs to specify the "Content-ID" of the object. If you have a requirement to be able to send a separate delivery rights object to re-enable the object, then you should manually define the "Content-ID" field.

-textXML - This optional parameter specifies that the rights object should be encoded in text XML format instead of the default binary WBXML format.

-out=outputFilename - This optional parameter specifies the output file name to contain the content with the DRM wrapper. If no filename is specified, DRMCOMP will default to the input objectFilename parameter, changing the file extension to ".dm".

Combined delivery DRM objects have a MIME type of "application/vnd.oma.drm.message". Note, however, that if you store a combined delivery file on a conventional web server, some mobile phones may not properly understand the content. The reason for this is because the "application/vnd.oma.drm.message" format uses MIME multipart encoding. The Content-Type header returned from the web server should indicate, Content-Type: application/vnd.oma.drm.message; boundary="drm-boundary". It is not possible to configure most web servers to return a boundary parameter. While the OMA specifications suggest that a device should be able to understand a DRM object even if the boundary parameter is not present in the Content-Type header, this could present potential problems.

For convenience, DRMCOMP uses a fixed boundary parameter of "--generic-drm-boundary". Therefore, the Content-Type header for any combined delivery DRM objects created by DRMCOMP should be:

```
Content-Type: application/vnd.oma.drm.message; boundary="--generic-
drm-boundary"
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

DRMCOMP – Separate Delivery

```
Usage: (Separate delivery)
   DRMCOMP objectFilename -separatedelivery [-headerFile=headerFilename]
                                           [-permission=play]
                                           [-permission=display]
                                           [-permission=execute]
                                           [-permission=print]
                                           [-limitCount=###]
                                           [-limitStartDate=yyyymmdd]
                                           [-limitEndDate=yyyymmdd]
                                           [-limitDays=###]
                                           [-contentType=type/subtype]
                                           [-contentID=cid]
                                           [-key=16Text-Or-32Hex-Characters]
                                           [-out=outputFilename]
                                           [-outRights=outputRightsFilename]
```

`objectFilename` - This is the name of the file that is to have the DRM wrapper applied.

`-separateDelivery` - This parameter indicates that a separate delivery DRM wrapper should be added to the message.

`-headerFile=headerFilename` - This optional parameter identifies a text header file that contains additional headers to be inserted into the header of the DRM object. *(See Section 5.2.4 of the DRM Content Format Version 1.0 specification.)* For example, a rights issuer URL, which is a URL that can be embedded into the content, which the user can connect to in order to renew access to the content, would be included in a such a rights header file as "Rights-Issuer: http://server:port/path/etc".

"DRM Permissions" specify what types of access are allowed against the objects in a message that is protected with DRM. The "permission=" parameters specify these permissions.

For example, an audio or video object requires "Play" permission before the user can access it. An image requires "Display" permission before the user can access it, and it requires "Print" permission if the user is to be allowed to print the image to a printer , perhaps over Bluetooth. An application requires "Execute" permission before the user can make use of the application. In all cases, the forward locking is assumed, so that the user is not allowed to forward or transfer the object from the device.

`-permission=play` - Enables "Play" permission for the protected object.

`-permission=display` - Enables "Display" permission for the protected object.

`-permission=execute` - Enables "Execute" permission for the protected object.

`-permission=print` - Enables "Print" permission for the protected object.

"DRM Constraints" specify constraints with regard to how long a DRM protected object object should remain accessible to the user.

It is possible to specify one or more of these "limit" constraints that follow.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

-limitCount=### - Specifies the the user can only access the DRM protected object this number of times before access is no longer allowed

-limitStartDate=yyyymmdd - Specifies that the user will not be allowed to access the DRM protected object until on or after the specified date.

-limitEndDate=yyyymmdd - Specifies that the user will not be allowed to access the DRM protected object after the specified date

-limitDays=### - Specifies that the user will be allowed to access the DRM protected object for this number of days after initial receipt of the object. The user can either enter a number of days here, or they can enter any valid value defined for the "<interval>" element in the OMA DRM Rights Expression Language specification. For example, P2Y10M15DT10H30M20S represents a duration of 2 years, 10 months, 15 days, 10 hours, 30 minutes and 20 seconds.

-contentType=type/subtype - This optional parameter defines the MIME content type associated with the object (e.g., image/jpeg, audio/mp3). If this parameter is not specified, DRMCOMP will attempt to identify the content type automatically based upon the file extension. To determine the file extension to MIME type mapping, DRMCOMP will consult the NowSMS MMSCTYPE.INI file, if present, and will then search the MIME file extension to content type database in the Windows registry.

-contentID=cid - This optional parameter defines a "Content-ID:" header to be included to identify the object. If this parameter is not present, DRMCOMP will generate one automatically. Should the object expire due to DRM constraints, it is possible to send a separate delivery rights object to re-enable this object, but the separate delivery rights object needs to specify the "Content-ID" of the object. NowSMS will include its dynamically generated "Content-ID" in the separate delivery rights object that it creates, but you must save this dynamically generated value if you wish to be able to send a later separate delivery rights object to re-enable an expired object.

-key=16Text-or32Hex-Characters - This optional parameter defines an encryption key to be used to encrypt the protected object. If this parameter is not present, DRMCOMP will generate one randomly. Note that the separate delivery rights object must include this encryption key. NowSMS will include its dynamically generated encryption key in the separate delivery rights object that it creates, but you must save this dynamically generated value if you wish to be able to send a later separate delivery rights object to re-enable an expired object.

-out=outputFilename - This optional parameter specifies the output file name to contain the content with the DRM wrapper. If no filename is specified, DRMCOMP will default to the input objectFilename parameter, changing the file extension to ".dcf".

-outRights=outputRightsFilename - This optional parameter specifies the output file name to contain the XML rights object for the protected content. If no filename is specified, DRMCOMP will default to the input objectFilename parameter, changing the file extension to ".dr".

Separate delivery DRM objects have a MIME type of "application/vnd.oma.drm.content".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The XML rights object may be sent via NowSMS using the **Send XML Settings** option in the web menu interface *(see page 118)*. The XML rights object can also be sent programmatically using HTTP POST. For more information, please see **Sending XML Settings Documents and Objects**, beginning on page 227.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Advanced Configuration Settings

There are some advanced configuration settings for NowSMS which can only be applied by directly editing the configuration files.

Please note that these settings are documented for use by advanced users only.

The primary configuration files for NowSMS are SMSGW.INI and MMSC.INI.

Most of the configuration settings that can be applied through the configuration program, are saved via configuration entries that are saved in these configuration files.

This section will not provide any documentation for settings that can be applied through the configuration program. It will only provide documentation for additional settings which can only be applied by directly editing the configuration files.

Please note that as these settings are not well documented, they do not receive the same amount of real-world testing. Use extreme care when applying any of these settings. You should always test thoroughly after applying any of these settings to make sure that NowSMS continues to function as you need it to.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

SMSGW.INI Parameters

SMSGW.INI, [SMSGW] section:

MMSDir=d:\path\

Specifies an alternate location for the "MMS-IN" subdirectory in which received MMS messages are deposited.

QDir=d:\path\

Specifies an alternate location for the "Q" subdirectory in which outbound SMS messages are queued. (Note: There was a bug in this setting for versions prior to v5.51 where the MMSC would not check this setting. The MMSC would queue messages into the default "Q" directory location, but they would never be sent.)

WAPPushInitiatorURI=http://server/

Set the default value for the initiator URI in WAP push messages generated by NowSMS. This value is displayed as the message sender with some phones, such as the SonyEricsson P900. (If an initiator URI is not present, some phones display the message as being from a blank or "unknown sender".) See **Technical Bulletin - WAP Push or OTA: Unknown Sender**.

WAPPushFlag=

Sets the default value for the "Push-Flag" header in a WAP push message. See **Technical Bulletin - WAP Push or OTA: Unknown Sender**.

MMSCOMPParameters=

Specify command line parameters to pass to MMSCOMP when messages are submitted via the web interface. This is most frequently used to specify the character set to be applied to text files that are submitted via the web interface.

NowSMS will automatically specify the UTF-8 character set, if necessary, for text posted in the "MMSText" variable, but it does not specify a character set for text that is submitted as a file upload.

To specify UTF-8 as the character set for text files submitted for MMS via the web interface, set MMSCOMPParameters=-cUTF-8

RetryDelay=
RetryDelayMultiplier=
RetryDelayAfterAttempts=
RetryDelayMax=
RetryMaxAttempts=

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The above parameters control retry behaviour for when NowSMS is submitting messages to an SMSC and an error occurs.

RetryDelay=#### specifies a number of seconds to wait to retry sending after an error condition, the default value is 30.

RetryDelayMultiplier=### specifies a multiplier to be applied for successive send failures, the default value is 1. For each failed attempt, the retry delay will be the product of RetryDelay*RetryDelayMultiplier*#FailedAttempts.

To use a fixed retry delay of RetryDelay, specify RetryDelayMultiplier=0.

RetryDelayAfterAttempts=### specifies that the retry delay should only be applied after ### failed attempts, the default value is 2. NowSMS will immediately retry a failed message send until it has made RetryDelayAfterAttempts, after which it will apply a retry delay.

RetryMaxAttempts=### specifies the maximum number of retries that NowSMS will attempt before a message is rejected, the default value is 20.

ReceiveSMSCharset=

When NowSMS calls a "2-way" command to pass details of a received SMS message, by default it will use the UTF-8 character set for the encoding of text messages.

This setting allows you to configure NowSMS to use a character set other than UTF-8 for transmitting text to a 2-way command. This value should specify a valid character set name (such as iso-8859-1 for the standard Western European character set, iso-8859-6 for Arabic, big5 or gb2312 for different Chinese encodings).

2WayProxy=

Specifies that HTTP requests defined in the "2-way" command table should be processed via an HTTP proxy server. Use a format of ip.address:port for specifying the address of the HTTP proxy server.

LogRetainDays=####

specifies the number of days for which logs (e.g., SMSOUT-yyyymmdd.LOG) should be retained. Values of 365 or larger are not supported.

Debug=Yes

Enables the SMSDEBUG.LOG file for troubleshooting.

DebugXmlSettings=Yes

Configures NowSMS not to delete temporary XML settings files that are generated when sending OTA configuration messages.

When sending OTA configuration messages, NowSMS generates an XML document containing the settings, before the settings are encoded for sending out over SMS. Normally, NowSMS deletes the XML documents as the settings are sent out. When this

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

setting is enabled, NowSMS leaves the XML documents behind in its "TEMP" subdirectory. This setting is intended primarily to assist users in creating their own XML settings documents. They can view the XML documents created by NowSMS via the menu interface, and then make custom modifications.

This setting should be removed from the INI file when it is no longer needed for testing.

OldModemHandler=Yes (obsolete)

This setting is referenced in various threads on the discussion board as a potential solution for dealing with problem modems. However, the setting is obsolete and should not be used with current versions of NowSMS.

ModemSendDelay=####

is a number of milliseconds (thousandths of a second) to pause in between sending commands to a GSM modem. Some poorly engineered modems might not be able to process modem commands as quickly as NowSMS sends them. This setting can force a short delay between commands.

SMSAccountingURL=

See **Technical Bulletin - Now SMS/MMS Accounting Callbacks**.

LogDirectory=d:\path

Specifies an alternate location for the log files (e.g., SMSOUT-yyyymmdd.LOG). Does not affect the location of debug log files.

BinaryDCS=

Specifies a hex value to be used as the DCS value when sending WAP Push messages or MMS notifications via WAP Push.

NowSMS default value is F5, however a value of 4 may be required on some networks.

LongTextDCS=

Specifies a hex value to be used as the DCS value when sending long text messages.

NowSMS default value is 0. (And we are not aware of any configurations where this had to be changed.)

ExtraHttpArgList=var1,var2,...

"var1", "var2", etc., are the names of HTTP variables. (Use only a comma to separate the variable names if you have multiple variables ... with no white space around the commas.) NowSMS parses requests that it receives, looking for these variable names. If the message is routed to an HTTP-based SMSC, it appends these variables and their associated values to the HTTP URL. (If the message is routed to any other type of SMSC, the variables are ignored.)

See http://www.nowsms.com/discus/messages/1/321.html.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

2WaySMSThreadCount=##

When NowSMS receives an SMS message over an SMSC connection, it can be configured to process a "2-way" command to process the message.

NowSMS queues the received SMS message, and a separate thread within NowSMS processes the callbacks, so other gateway activity of sending and receiving messages can continue while the callbacks are being processed.

This approach is reasonably efficient. However, if you receive a large volume of messages, our experience has shown that HTTP callbacks can only be processed at a rate of 3 to 4 messages per second.

If you need to process inbound messages at a faster rate, it is possible to allocate additional threads to the receive SMS processing. This parameter specifies how many threads should be allocated for processing received messages.

In practice, the overhead of allocating additional threads for this processing is relatively low, as the threads spend most of their time waiting for a message to process or waiting for an HTTP response from the callback that is issued.

SMPPRejectErrorCodes=x,y,z

(v5.51g+) This option specifies SMPP error codes that should be considered a permanent error (for which NowSMS will not retry sending the message). By default, NowSMS treats the following SMPP error codes as a permanent error: A (ESME_RINVSRCADR - invalid source address), B (ESME_RINVDSTADR - invalid destination address), 66 (ESME_RX_R_APPN - ESME Receiver Reject Message). To add additional error codes that should be considered as a permanent error, this setting can be used. The value for this setting can contain a comma delimited list of SMPP error codes. Values should be specified in hex, with no leading zeroes, e.g., "SMPPRejectErrorCodes=A,B,66".

MaxSMPPShortCodeLen=#

(v5.51g+) By default, a source_addr_ton (source address type of number) value of 3 will be used for any messages where the sender address is numeric, and of length less than or equal to 5 digits. To disable this behaviour, use the setting MaxSMPPShortCodeLen=0. (Note: If the sender address contains alpha characters, source_addr_ton will be 5 by default. If the sender address starts with a "+", then source_addr_ton will be set 1 by default. In all other cases, the default value for source_addr_ton will be 0.)

SMPPThrottleErrorDelay=##

(v5.51b+) If an SMPP SMSC returns a throttling error (ESME_RTHROTTLED), previous versions of NowSMS would delay 15 seconds before attempting to process another message over that SMPP connection. This default delay was changed to 5 seconds in v5.51b. It has been observed that some SMSCs might return this error condition when it will not accept any more messages for a particular phone number, in which case a general delay is not desirable. To change this delay, use this parameter, where ## is a number of seconds. SMPPThrottleErrorDelay=0 will disable the delay.

InitRetryDelay=

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

InitRetryDelayMultiplier=
InitRetryDelayAfterAttempts=
InitRetryDelayMax=

(v5.51b+) By default, if NowSMS cannot connect to an SMSC (or initialise a modem), it will wait 20 seconds before attempting another connection. For each successive connection failure, NowSMS will wait an additional 20 seconds (e.g., 40 seconds after 2 failures, 60 seconds after 3 failures, etc.), up to a maximum of 300 seconds (5 minutes) between retries. The reason for these delays is to adhere to various operator guidelines which are designed to prevent an SMSC from becoming overloaded with reconnections when recovering from a failure. It is possible to modify these retry timings using the following SMSGW.INI file parameters under the [SMSGW] section header:

InitRetryDelay=#### specifies a number of seconds to wait to retry after a connection failure, the default value is 20.

InitRetryDelayMultiplier=### specifies a multiplier to be applied for successive connection failures, the default value is 1. For each failed attempt, the retry delay will be the product of InitRetryDelay+((InitRetryDelay-1)*InitRetryDelayMultiplier*#FailedAttempts). To use a fixed retry delay of InitRetryDelay, specify InitRetryDelayMultiplier=0.

InitRetryDelayAfterAttempts=### specifies that the retry delay should only be applied after ### failed attempts, the default value is 1.

InitRetryDelayMax=### specifies the maximum number of seconds that NowSMS will allow to elapse between connection retries, the default value is 300.

WAPPushIncludeDate=Yes

(v5.51a+) Some Samsung phones display a date/time associated with received WAP Push messages, where the date/time displayed is inaccurate. This setting forces NowSMS to insert a date/time stamp into WAP Push messages. Note that this setting was implemented to resolve a specific incident report, and has not been tested against a wide range of other phones.

ExcludeSmsDetailsFromLog=

(v5.51a+) In standard operation, NowSMS logs the text of sent messages in the SMSOUT log files. This setting prevents the SMS message details (text) from being included in the standard log files.

MaxRingToneFragmentCount=##

(v2006+) The "Send EMS Ringtone" function will reject sending a ringtone if it would result in more than this number of SMS fragments being sent. The default value is 6.

OldModemReceiptFormat=Yes

(v2006+) Beginning with NowSMS 2006, delivery receipt messages received over a GSM modem SMSC connection are translated into a format that is compatible with SMPP implementations (message text starts with "id:"). Use this setting to cause NowSMS to revert to the receipt format used in previous versions (message text starts with "Report:").

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

DebugMaxSize=####

(v2006+) This setting specifies the maximum size in MB of the SMSDEBUG.LOG, when it the debug log is enabled.

SMPPServerEnquireLink=###

(v2006+) Specifies an enquire link (idle) timeout in seconds for the SMPP server. If the SMPP server does not receive an enquire link (or other command) within this timeout period, the connection will be automatically terminated. The default setting is 120 seconds. A value of 0 can be used to disable this timeout.

2WayReplySameServer=Yes

(v2008.03.23+) This setting specifies that replies from all 2-way commands should be routed back via the same SMSC connection as which the original message was received.

2WayReplyCopySMPPOptions=xxxx,yyyy

(v2008.11.05+) This setting can contain a comma delimited list of "SMPPOptions" TLV parameters that should be automatically copied from a received message to the reply message generated by the 2-way command. Any options specified in this list must be included in the [SMPPOptions] section of SMSGW.INI.

2WayKeepAlive=No

(v2008.11.05+) For improved performance, keep-alive sockets are enabled by default for HTTP-based 2-way SMS commands. This setting is used to disable the use of keep-alive sockets for 2-way command processing if there is a problem with keep-alive sockets in a particular environment.

AccountingKeepAlive=No

(v2008.11.05+) For improved performance, keep-alive sockets are enabled by default for SMS accounting callbacks (SMSAccountingURL). This setting is used to disable the use of keep-alive sockets for SMS Accounting callbacks if there is a problem with keep-alive sockets in a particular environment.

SMSGW.INI, section specific to an SMSC definition:

SMSCSendLimit=

SMSCSendLimit=x/y setting can specify that the gateway will send no more than x messages per y seconds. If y is not specified, then the default is 1. For example, to limit a connection to 1 message every 5 seconds, specify SMSCSendLimit=1/5. To limit a connection to 3 messages per second, specify SMSCSendLimit=3 or SMSCSendLimit=3/1.

Charset= (HTTP SMSC definitions only)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Specifies the character set to be used when submitting text messages to an HTTP based SMSC. By default NowSMS uses UTF-8, however some service European or American providers might expect iso-8859-1 (the standard Western European "text" character set).

ExcludeSMSC=Yes (GSM modems only)

Necessary when using the Siemens M1 as a modem. This older modem does not properly implement the syntax for sending SMS messages in PDU format, and does not expect an SMSC field to be specified in the PDU. This parameter allows NowSMS to be used with the Siemens M1. Without this setting the M1 will report "ERROR" on every message send attempt.

SMSC= (GSM modems only)

When sending messages via a GSM modem, by default NowSMS specifies that the default SMSC should be used. This default SMSC value is normally pre-configured on the SIM card.

Some older GSM modems may require the SMSC value to be explicitly set when NowSMS sends a message. (We have only observed this once with an old Ericsson DI28 modem, which is an external snap on infrared modem that attaches to some older Ericsson phones.)

This SMSC address varies depending upon what operator you are subscribed to.

You can set the SMSC address by manually editing the SMSGW.INI file. Under the section header for the modem configuration (e.g., [Modem - ...], add SMSC=+phonenumber, where "+phonenumber" is the address of the SMSC. Here's a link to a good list of SMSC phone numbers: http://www.cellular.co.za/smsc_lists.htm. A more reliable way of determining the correct SMSC number is to put your SIM into a phone, and go through the SMS configuration menus on the phone to determine the currently configured SMSC number. When you enter the SMSC phone number, always start it with a "+" and don't include any other non-numeric characters (no dashes or dots) in the address.

DelayAfterSend= (GSM modems only)
DelayAfterError= (GSM modems only)

This setting can be useful for modems that encounter problems with the speed at which NowSMS attempts to send messages. In the appropriate [Modem Name] section of the SMSGW.INI file, the following settings can be specified: DelayAfterSend=#### specifies a number of milliseconds to wait after successfully sending a message (1000 milliseconds = 1 second). DelayAfterError=#### specifies a number of milliseconds to wait after a message send attempt fails.

CommandPreInit#= (GSM modems only)
CommandPostInit#= (GSM modems only)
CommandPreSend#= (GSM modems only)
CommandPostSend#= (GSM modems only)

Substitute # with a number, starting at 1, and increasing sequentially for each additional command to be sent. CommandPreInit is sent before NowSMS performs its modem initialisation sequence. CommandPostInit is sent after NowSMS performs its modem initialisation sequence. CommandPreSend is sent before NowSMS attempts to send a message using the modem. CommandPostSend is sent after NowSMS attempts to send a message using the modem. Examples:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

CommandPreInit1=ATZ
CommandPreInit2=ATE0V1
CommandPreInit3=AT+CSMS=0

RouteName=xxxxx

(v5.51b+) When submitting a message via HTTP, it is possible to indicate that the message should be routed via a specific SMSC. The HTTP interface supports "&SMSCRoute=xxxxx", where the value of this setting can be the name of a defined SMSC (e.g., "Bluetooth Modem" or "SMPP - a.b.c.d:xyz"), or it can be a route name that is defined to be associated with one or more SMSCs. This parameter setting is used to define a route name to be associated with an SMSC.

PriorityFlag=#

(v5.51a+) For SMPP connections, this setting can specify the priority_flag value to use when submitting messages. This flag will apply to all messages submitted via a specific SMSC link.

AllowedUserOnly=
AllowedUser1=
AllowedUser2=
AllowedUser...=

(v5.51a+) These configuration settings are defined to limit particular outbound "SMSC" connections to use by selected "SMS Users" accounts.

AllowedUserOnly=Yes -- This setting specifies that the SMSC connection can only be used by users that are specifically authorised to use this SMSC. When routing messages from users that are not specifically authorised for this connection, NowSMS will not use this SMSC and route messages as if this SMSC definition did not exist.

AllowedUser1=username -- This setting can be repeated with sequentially assigned numbers (e.g., AllowedUser2=xxx; AllowedUser3=yyyy). This entry is used to specify specific "SMS Users" accounts that are allowed to use the SMSC connection when AllowedUserOnly=Yes.

Note: It is NOT necessary to restart NowSMS when making updates to the AllowedUserOnly or AllowedUser1 settings in SMSGW.INI. These changes will be detected dynamically by NowSMS.

InternationalPrefix=###

(UCP/EMI only) If a destination phone number is of international format, the "+" character will be replaced with the prefix specified here. (This setting is only respected if the sender address is NOT in international format.)

CommandTimeout=###

(SMPP only) Specifies a timeout in seconds to wait for an SMPP response. Default is 120 seconds.

NullRemap=40

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

(SMPP only) This setting is used to resolve character set conversion problems where the "@" character is not handled correctly on received messages. Note that the preferred solution for this problem is to define the "SMSC Character Set" as "iso 8859-1 (Latin)" under "Advanced Settings". This setting should only be used if problems persist after configuring that setting.

Validity=##D (or ##H or ##M)

(v2006+) *Supported by outbound GSM modem and SMPP SMSC connections only*. This parameter specifies a default validity period for all messages sent via this SMSC connection, as an interval defined in hours, minutes or days. If the SMSC cannot deliver the message within the specified validity period, the SMSC is instructed to discard the message. *(Note that this setting is not supported by all SMSCs.)* Specify ##D for a validity period in days, ##H for a validity period in hours, or ##M for a validity period in minutes, where ## is a numeric value *(e.g., 30D for 30 days or 7H for 7 hours)*.

BinaryOrTextOnly=Text (or Binary)

(v2006+) Indicates that this SMSC connection should only be used for text or binary messages, as indicated. If this parameter is set to "Text", NowSMS will not try to route any binary messages via this connection. Similarly, if this parameter is set to "Binary", NowSMS will not try to route any text only messages via this connection. The intent is that some SMSC connections may be a cheaper alternative for sending text only messages.

InsertSenderAsCallbackNum=Yes

(v2006+) SMPP Only - When this parameter is set, all messages that are sent via this SMSC link have the callback_num field inserted, with a value that matches the message sender.

GenerateUserMessageReference=Yes

(v2007.06.27+) SMPP Only – This parameter causes NowSMS to include an incrementing user_message_reference value with each message submission to the SMSC. This special configuration setting was required for a customer connecting to Movistar Peru.

BackupForRoute=xxxxxx

(v2008.12.04+) Defines this SMSC connection as a backup (fail-over) route for another SMSC connection. A backup route is activated only when a primary route is down. To configure a backup route, define the SMSC connection to NowSMS as normal, then manually edit the SMSGW.INI file. Under the SMSC section header (e.g., [Modem - driver name] or [SMPP - server:port]), add BackupForRoute=xxxxxx where xxxxxx is either the name of another SMSC connection or a "RouteName=" setting defined for one or more SMSC connections. If xxxxxx is the name of another SMSC connection, this SMSC connection will only be activated if that other SMSC connection is down. If xxxxxx is a "RouteName=" setting defined for one or more SMSC connections, this SMSC connection will only be activated if all other SMSC connections with this route name are down.

TextDCS=##
BinaryDCS=##
UnicodeDCS=##

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

(v2009-06-30+) (SMPP only) ## is a hex value to be used as an override for the data_coding value all 7-bit text, binary and/or Unicode messages. For example, the SMPP specification defines data_coding values of 2 and 4 for 8-bit binary messages. When NowSMS receives a message via SMPP with a data_coding value of 2, it automatically remaps this value to 4 for compatibility with the DCS (data coding scheme) value used in GSM environments. Some SMSC implementations expect the data_coding value for binary messages to be 2. Using this setting with BinaryDCS=2 for a specific SMPP connection accomplishes this configuration requirement. Another example is USSD environments, where 0x0F is often the preferred DCS value for USSD text messages, and 0x48 is the preferred DCS for USSD unicode messages. When an override is applied, all messages of the specified type (text, binary or Unicode) will have their DCS value converted to the override value.

IntermediateNotification=Yes

(v2009-06-30+) (SMPP only) When this setting is enabled, the "intermediate notification" bit will be set in the registered_delivery flag any time a delivery receipt is requested. If a delivery receipt is not requested, this bit will not be set.

ReceivePoll=##

(v2009.06.30+) (GSM Modem Only) This setting specifies the interval in seconds between polling attempts when NowSMS polls the GSM modem to see if any new messages have been received. By default, the polling interval is set to 10 seconds when the "SMS Message Storage" option is set to "Direct to Modem". The polling interval defaults to 2 seconds for other "SMS Message Storage" settings. (In previous versions of NowSMS, the polling interval was always 1 second.)

OldPollingLogic=Yes

(v2009.06.30+) (GSM Modem Only) Changes were made in how NowSMS polls modems for received messages in this release. This setting causes NowSMS to use the polling logic from older releases.

SMSGW.INI, [SMPPOptions] section:

This section is used to enable support for SMPP Optional parameters. In particular, mBlox customers may wish to configure these settings in order to enable support for mBlox specific settings related to premium rate SMS.

Other SMS providers might also use SMPP Optional parameters for similar purposes.

The [SMPPOptions] section contains a list of SMPP optional parameters that NowSMS should support.

Some optional parameters are defined in the SMPP specification, while others, such as the mBlox settings that we reference below, are vendor specific.

The format of entries in the [SMPPOptions] section is:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

setting_name=SMPPTagValue,Type,Len

setting_name is a descriptive name for the setting. For each setting that is defined, NowSMS will support an additional parameters when submitting a message to NowSMS via an HTTP URL request. "&SMPPOption_setting_name=" will specify a value to be set for this parameter.

For each optional parameter that is defined, NowSMS will also route these parameters to HTTP-based 2-way commands. If a message is received which contains values for any optional parameter defined in the [SMPPOptions] section, NowSMS will automatically append "&SMPPOption_setting_name=value" to the 2-way URL, if the optional parameter is present in a received parameter. It is not necessary to add any variables to the 2-way command template, as these values will be appended automatically if present in a received message.

SMPPTagValue is the SMPP tag value associated with the parameter, in hex format. For example, the SMPP v3.4 specification defines the tag value for user_message_reference to be 0204.

Type can be either String, CString or Integer. String corresponds to the "Octet String" definition in the SMPP specification. CString corresponds to the "C-Octet String" (null terminated) definition in the SMPP specification. Integer corresponds to the "Integer" definition in the SMPP specification.

Length specifies the field length, and is optional. If specified for a String parameter, it specifies a fixed length for the string (longer strings will be truncated, shorter strings will be padded with nulls to meet this length). If specified for an Integer parameter, the value should be either 1 or 2, otherwise NowSMS will choose a size based upon whether the value requires 1 or 2 bytes to encode. Length is ignored for CString typed parameters.

In the context of mBlox, the following settings are intended to enable support for their premium rate parameters:

[SMPPOptions]
mblox_operator=1402,String,5
mblox_tariff=1403,String,5
mblox_sessionid=1404,String,45
user_message_reference=204,Integer,2

When these settings are present, additional parameters are supported when submitting a message to NowSMS via an HTTP URL request. "&SMPPOption_mblox_operator=" will specify a value for the destination operator. "&SMPPOption_mblox_tariff=" will specify a value for the premium rate tariff associated with the message. "&SMPPOption_mblox_sessionid=" is required for some premium rate SMS operator implementations. For additional information on any of these parameters, please refer to the mBlox SMPP Gateway Version 3.0 Manual.

"&SMPPOption_user_message_reference=" is a generic option that allows you to set/retrieve the value of the SMPP "user_message_reference" variable. An mBlox FAQ suggests that it is possible to use this option to specify a billing reference.

Please note that mBlox support will often suggest that you use a value of "0" for the mblox_tariff initially. However, customers have reported to us that mBlox appears to require a 5 digit code for this field, and that "00000" should be used instead of "0".

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

When these SMSGW.INI file settings are present, NowSMS will also route these parameters to HTTP-based 2-way commands. If a message is received which contains values for either of these settings, NowSMS will automatically append "&SMPPOption_mblox_operator=value" and/or "&SMPPOption_mblox_tariff=" to the 2-way URL. It is not necessary to add any variables to the 2-way command template, as these values will be appended automatically if present in a received message.

In addition to supporting these options via HTTP URL request, it is possible to configure default settings for any of these options for each outbound SMPP connection. To define default settings, manually edit the SMSGW.INI, and in the section header for an SMPP connection (e.g., [SMPP - ip.address:port]), add a "DefaultSMPPOptions=" setting, where the value of this setting can contain any of the "SMPPOptions" settings. For example, "DefaultSMPPOptions=mblox_tariff=00000". To include multiple options, separate the entries with a ";", for example, "DefaultSMPPOptions=mblox_tariff=00000;user_message_reference=1".

SMSGW.INI, [SMPP] section:

DeliveryReceiptFlag=Yes

(v2006+) Forces a delivery receipt to be requested for every message that is sent out through any SMPP connection.

DisableDeliveryReceipt=Yes

(v2006+) Disables delivery receipt requests for every message that is sent out through any SMPP connection.

SMSCRouteTLV=####

(v2009+) When this configuration parameter is present, NowSMS enables a TLV parameter matching the #### hexadecimal tag number. When NowSMS routes a a message to an SMPP client, this TLV parameter will contain the route information from which the message was received (either the SMSC host name and port or the "RouteName=" parameter configured in the SMSGW.INI for the SMSC). Similarly, an SMPP client can specify this parameter when submitting a message to request a specific outbound route for the message. The behaviour of this parameter is similar to the "SMSCRoute=" parameter that is supported by the NowSMS HTTP interface. Additional discussion of this feature can be found at http://www.nowsms.com/discus/messages/1/24485.html.

SMSGW.INI, [UCP] section:

IncludeXSerDCS=No

By default, any non-text DCS (data coding scheme) value will be encoded in the XSer field. Use this setting to disable this behaviour.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

LongSMSTextEncoding=Yes

This field specifies that long text messages should be encoded in text format. By default, NowSMS will convert long text messages to a binary format with UDH before sending. If long text messages are garbled, try this setting.

MMSC.INI Parameters

MMSC.INI, [MMSC] section:

Debug=Yes

Enables the MMSCDEBUG.LOG file for troubleshooting.

DebugLogDirectory=d:\path

Specifies an alternate location for the MMSCDEBUG.LOG file when Debug=Yes is configured.

LogDirectory=d:\path

Specifies an alternate location for the log files (e.g., MMSC-yyyymmdd.LOG). Does not affect the location of debug log files.

DataDir=d:\path

Specifies an alternate location for the MMSCDATA directory, where MMS messages are stored. By default, this is a directory named MMSCDATA beneath the NowSMS installation.

VASPQDir=d:\path

(v2006+) Specifies an alternate location for the VASPQ directory, where MMS messages to be routed to an external MMSC connection are stored. By default, this is a directory named VASPQ beneath the NowSMS installation.

DataRetainDays=xxx

xxx is the number of days that messages should be retained. The default is 30 days. Note that this setting has not been tested for values exceeding one year, but a value of 180 days (approx 6 months) should be fine.

LogRetainDays=####

specifies the number of days for which logs (e.g., MMSC-yyyymmdd.LOG) should be retained. Values of 365 or larger are not supported.

ExpireDynamicLinks=###

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

This configuration parameter is used to delete dynamically generated links (such as those used when sending MMS or Multimedia WAP Push), after they are accessed. ### is the number of minutes after the link is first accessed before it should be automatically deleted

AdminPasswordDelete=xxxxxxxx

Allows a separate password to be required for web administration access before a user account can be deleted via the web interface.

AdminPasswordModify=xxxxxxxx

Allows a separate password to be required for web administration access before a user account can be modified via the web interface. When this setting is present, the regular admin password can only be used to view account details.

UAProfProxy=ip.address:port

(v5.51b+) Specifies that the MMSC should download UAProf files via an HTTP Proxy server (used when MMSC does not have direct internet connectivity). This setting must specify an IP address and port number for the HTTP Proxy.

MMSNoSubject=subject line text

(v5.51b+) Some operator MMSCs do not like a blank subject header. When routing a message from SMTP to MMS, if there is no subject header, or the subject header is blank, NowSMS does not include a subject header in the resulting MMS message. To include a subject, use this setting to specify the text to be inserted as the subject when a subject header is not present in a message received from SMTP.

DomainNameAlias1=domaina.com
DomainNameAlias2=domainb.com
DomainNameAlias3=domainc.com
DomainNameAlias4=domaind.com
DomainNameAlias5=domaine.com
DomainNameAlias6=domainf.com
DomainNameAlias7=domaing.com
DomainNameAlias8=domainh.com
DomainNameAlias9=domaini.com
DomainNameAlias10=domainj.com

These settings are used to specify alternate domain names for which the MMSC's SMTP server will accept e-mail for routing to MMS recipients. Normally, the MMSC will only accept messages addressed to username@domain, where domain matches either the "Domain Name for MMS E-Mail" or "Local Host Name or IP Address" setting.

SMSDomainNameAlias1=domaina.com
SMSDomainNameAlias2=domainb.com
SMSDomainNameAlias3=domainc.com
SMSDomainNameAlias4=domaind.com
SMSDomainNameAlias5=domaine.com
SMSDomainNameAlias6=domainf.com
SMSDomainNameAlias7=domaing.com

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

SMSDomainNameAlias8=domainh.com
SMSDomainNameAlias9=domaini.com
SMSDomainNameAlias10=domainj.com

These settings are used to specify alternate domain names for which the MMSC's SMTP server will accept e-mail for routing to SMS recipients. Normally, the MMSC will only accept messages for routing to SMS when addressed to username@domain, where domain matches the "Domain Name for SMS E-Mail" setting.

VASPQRetryMaxAttempts=##

(v5.51b+) This configuration option limits the number of retries for processing of messages in the MMS Outbound Routing queue, which contains messages that are being routed to an external MMSC connection (EAIF, MM1, MM4, MM7). By default, retries will be performed for 24 hours, with the retry interval moving from 1 minute to 2 minutes to 3 minutes, up to 15 minutes per retry. This setting will limit the number of retries.

More Discussion at http://www.nowsms.com/discus/messages/485/8212.html

HeaderUserAgent=any text

This setting specifies a value to be set for the "User-Agent:" header when submitting/retrieving MMS messages from/to an operator MMSC.

HeaderProfile=http://host.name/path/filename.ext

This setting specifies a value to be set for the "Profile:" header when submitting/retrieving MMS messages from/to an operator MMSC. This should be a URL that points to a User Agent profile (UAProf).

UndeliverableRouteToSMS=VASPOutboundRouteName

(v5.51d+) Specifies the name of an MMSC Outbound Route that is defined in the "MMSC Routing" list, which must be of the type "Convert to SMS with Web Link". By default, if an MMS message has not been retrieved within 120 minutes, the message will be rerouted to be sent as an SMS with a web link for accessing the MMS content.

UndeliverableRouteToSMSTimeout=####

(v5.51d+)#### is a value in minutes that changes the time period after which the UnderliverableRouteToSMS setting is applied.

Nokia3510Compatible=No

This setting enables a more compact form of encoding for MMS notifications, as it uses a single byte for encoding the MMS content type in the MMS notification. The default setting requires 33 bytes for encoding the content type, so this setting will save 32 bytes in the size of the MMS notification message that is sent over SMS. However, some early MMS phones did not understand the short encoding for the MMS content type, and this setting will cause those phones to be unable to receive any MMS messages. It has been observed that the original releases of the Nokia 3510 and Panasonic GD87 did not understand the short encoding, and other phones may also be affected. Use this setting with caution.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSNotificationNoSubject=Yes

This setting will disable the inclusion of the subject header in the MMS notification message delivered over SMS, resulting in a smaller size for the MMS notification message. The only known side effect of this setting is that if a user has configured their phone to prompt them before downloading an MMS message, the user will be unable to see a message subject until after the message is downloaded.

MMSNotificationNoSender=Yes

This setting will disable the inclusion of the sender header in the MMS notification message delivered over SMS, resulting in a smaller size for the MMS notification message. Use of this setting is not recommended. Some phones, in their default configuration, will ignore MMS notifications where there is no sender specified. Additionally, some SonyEricsson phones will disable reply capability when this setting is enabled, even though the sender will clearly be displayed in the received message.

CompactMMSURL=Yes

This setting uses a more compact encoding of the URL value that is sent out in MMS notifications, saving several bytes in the size of the MMS notification that is sent out over SMS.

EnableReadReceipt=No

(v5.51d+) This setting removes the "read receipt request" attribute from any messages that are submitted to the MMSC by local users (and MM1 or EAIF VASPs).

MMSMessageSizeLimit=#####

Specifies a maximum size, in KB, for MMS messages that will be accepted by the MMSC when submitted by local users (and MM1 or EAIF VASPs).

NoChargeMMSToEMail=Yes

When this setting is present, MMS messages sent to e-mail recipients will not count against the daily or monthly quotas.

MM4AckToSourceIP=Yes

When this setting is present, MM4 ACKs will always be sent back to the same IP address from which the MM4 message submission was received. Usage of this setting is NOT recommend. Instead, you should define an MM4 outbound route under "MMSC Routing", and configure the "MMSC VASP" account to route MM4 ACKs via the appropriate route.

ReplaceCIDInSMIL=Yes

When this setting is present (and content conversion is enabled), content conversion will be forced upon any messages where the SMIL file contains "cid:" (content-id) references in the SMIL. Content conversion always converts these references to point to "Content-location" references instead. There is no known reason to use this setting.

RemovePlusFromEMail=Yes

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

When this setting is present, the leading "+" character will be stripped from phone numbers when routing from MMS to e-mail, so that a "+" character is not included as the sender address in the outbound message.

EnableKeepAlive=No

This setting disables the use of keep-alive sockets by the MMSC. By default, the MMSC will enable the use of keep-alive sockets when communicating with MMS clients that request them through HTTP/1.1. When this setting is used, the MMSC will inform the clients that keep alive sockets are not supported. (This setting is not recommended, as it has been observed that some MMS clients assume that keep-alive sockets are supported, and will report an error after downloading a message if keep-alive sockets are disabled.)

EMailAutoSmil=No

By default, NowSMS will automatically generate a SMIL presentation for content that is received via SMTP without an existing SMIL presentation. This setting disables that behaviour.

EMailAutoSmilOldFormat=Yes

(v5.51c+) Beginning in v5.51c, the way that SMIL that is auto-generated when routing e-mail to MMS was changed to include a simple "layout" section, and "region" attributes for included content. This is necessary for proper message display on Blackberry devices. If this change causes a problem when upgrading from a prior version, this setting will revert to the old auto-generated SMIL format.

EMailKeepSmil=Yes

(v5.51c+) Beginning in v5.51c, when an MMS message is routed to an e-mail recipient, any SMIL presentation attachment is discarded, as this can cause confusion for e-mail recipients who are not familiar with SMIL. Use this setting to enable the SMIL to be retained.

EMailAutoSmilParDur=xxxxx

When automatically generating SMIL for SMTP e-mail messages routed to MMS recipients, an option has been added to specify that a "dur=" parameter be included in the paragraph headers. By default, if the message only requires a single page (paragraph element), then no "dur=" parameter will be specified. If a message requires multiple pages, a "dur='5000ms'" parameter will be included in the paragraph header. However, these settings may not be appropriate for all environments. Additionally, there is a bug in some versions of the SonyEricsson P900 where messages will display as blank if a paragraph header does not have a "dur=" parameter. To override the default behaviour, use this parameter, where "xxxxx" specifies a duration in milliseconds (ms) or seconds (s) such as "5000ms" (5000 milliseconds = 5 seconds) or "30s" (30 seconds).

EMailSMSTextOnly=Yes

This setting will suppress the SMTP sender and subject from being included when routing SMTP e-mail to SMS recipients. Only message text from the e-mail will be included in the

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

SMS. This setting is useful when SMTP is being used as an API for submitting SMS messages, instead of as a more general e-mail to SMS gateway.

DisableMMSDirectDelivery=Yes

(v5.51a+) This setting is used when NowSMS is primarily being used for e-mail to MMS purposes, and the MMS connection is via an external MMSC connection instead of MMS direct delivery. This setting will stop NowSMS from performing MMS direct delivery for accounts that are defined in the "MMSC Users" list. By default, NowSMS will always perform MMS direct delivery to accounts that are defined in the "MMSC Users" list. When NowSMS is being used for e-mail to MMS, this setting allows those recipients to be routed via an external MMSC connection

DisableTransactionID=Yes

(v5.51b+) This setting is useful when NowSMS is being used as a direct delivery MMSC, but not as the user's primary MMSC. This setting causes the MMSC not to request any delivery acknowledgments from the receiving MMS client. These acknowledgments are always posted to the MMSC that is configured as the primary MMSC in the phone, so if NowSMS is not the primary MMSC, these acknowledgments are posted to the wrong MMSC.

MSISDNHeader=HTTP-Header-Name

This setting is used to identify the phone number (MSISDN) of MMS clients when they submit messages to the MMSC. This setting specifies the name of the HTTP header that will contain the MSISDN on MMS submission requests from MMS clients. This header is usually inserted by the WAP proxy. For example, with NowWAP, the appropriate setting would be: MSISDNHeader=X-MSISDN

See http://www.nowsms.com/support/bulletins/tb-nowsms-002.htm for more general detail about this capability.

This setting can support a comma-delimited list of headers, so that the MMSC can be configured to accept MSISDN information from one of multiple headers, with the first listed header always being given priority.

MSISDNHeaderGateways=ip.address

This setting specifies a list of one or more IP addresses from which the MMSC will accept the MSISDNHeader. This is to prevent forged requests, where another gateway or application inserts an MSISDN header to attempt to fool the MMSC. One or more IP addresses can be listed in this configuration setting. Each address must be separated by a comma. Wildcard addresses are supported by placing a "*" in place of a position within the IP address. For example, a setting of 9.10.11.* would mean that the MSISDNHeader would be accepted from any request originating from an IP address in the range of 9.10.11.1 thru 9.10.11.255.

MSISDNHeaderPrefixConvert=xxx:yyy

(v5.51a+) This setting is used for applying a standard conversion to MSISDN header from a WAP proxy. MSISDNHeaderPrefixConvert=xxx:yyy allows you to convert all MSISDN values reported by the WAP proxy which start with "xxx" so that they start with "yyy" instead.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Multiple conversion entries can be specified by including them in the header, separated by a comma (e.g., xxxx:yyyy,aa:bb).

MSISDNHeaderDefaultCountryCode=##

This setting specifies a default country code to be applied to MSISDN numbers presented in the MSISDNHeader, so that the gateway can convert the MSISDN address to international format automatically.

MSISDNHeaderLocalPrefix=#

This setting specifies the default prefix that is used for phone numbers in the MSISDN header that are in local (national) format. For example, in the UK, the MSISDNHeaderDefaultCountryCode would be 44, and the MSISDNHeaderLocalPrefix would be 0. With these settings, and MSISDN header of "07778001210" would automatically be converted to "+447778001210" by the MMSC

MSISDNHeaderAutoProvision=Yes

This setting specifies whether or not user accounts should be automatically provisioned on the MMSC the first time that a user sends a message through the MMSC. This removes the requirement to automatically provision accounts. Any user that makes a request through the appropriate WAP gateway with the MSISDNHeader set will be automatically provisioned on the MMSC.

MSISDNHeaderEricssonFormat=Yes

This setting specifies that the MSISDN header information is in a format typically used by an Ericsson WAP Gateway.

MSISDNHeaderNokiaFormat=Yes

This setting specifies that the MSISDN header information is in a format typically used by an Nokia WAP Gateway.

MSISDNHeaderConvertToLocalNumber=Yes

When MSISDNHeaderDefaultCountryCode and MSISDNHeaderLocalPrefix are enabled, NowSMS will convert MMS sender and recipient information to international number format when delivering messages. As many users may be unfamiliar with international number format, this setting translates the phone number format that appears in the MMS message so that it appears in local number instead.

MSISDNHeaderRequiredForReceive=Yes

(v5.51c+) When this setting is present, in order for a client to retrieve an MMS message from the MMSC, the MMSC will check to make sure that the retrieving client presents an MSISDNHeader that matches the intended message recipient.

MSISDNMatchRequiredForReceive=No

(v5.51c+) Similar to MSISDNHeaderRequiredForReceive explanation. By default, NowSMS will expect a URL request to include the appropriate MMS filename, and the phone number

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

of an intended recipient. The MMS message can only be downloaded if the MMS file exists, and the phone number in the URL matches an intended recipient. This check is performed by default to make it considerably more difficult for an automated process to guess at MMS filenames. If there is a reason that this check needs to be disabled, use this setting.

MSISDNHeaderDNCode=
MSISDNHeaderDNOffsetMin=
MSISDNHeaderDNOffsetMax=

These settings all are used in conjunction with MSISDN parsing in Argentina, where the local number format includes a DN code in the middle of the phone number string. Further information can be supplied on request. Use these settings in Argentina:

MSISDNHeaderDefaultCountryCode=54
MSISDNHeaderDNCode=15
MSISDNHeaderDNOffsetMin=3
MSISDNHeaderDNOffsetMax=6

SMTPIPAddressList=ip.address.1,ip.address.2,192.168.1.*

This parameter can have a comma-delimited list of IP addresses (no white space, if you specify multiple addresses, separate with a comma only) that are allowed to connect via SMTP and send to *any* phone number using an addressing format of phonenumber@domain.name.sms.or.mms. The IP addresses specified, can also include a "*" place holder as a wildcard match for all addresses in that network (for example, *.*.*.* would open access to any address, so you could use that value if you have complete confidence in your firewall).

MMSCIPRestrict=ip.address.1,ip.address.2,192.168.1.*

This parameter restricts connections to the MMSC to a specified list of IP addresses. This setting can contain a a comma delimited list of IP addresses that are allowed access to the MMSC. Do not include any spaces between addresses. To specify a wildcard address for a subnet, use the "*" character (for example, 192.168.1.* to allow all addresses in the 192.168.1 subnet).

LocalNumberPrefix=xxxxx
LocalNumberMaxLength=#

These settings are used in situations where a provider serves a particular city code within a country code, which sometimes happens with island operators.

If the MMSC receives a message addressed to a recipient whose phone number length is less than or equal to LocalNumberMaxLength, it automatically prepends LocalNumberPrefix to the number, before applying standard local number to international number conversion.

ShortCodeMaxLength=####

Specifies the maximum length allowed for short codes. When the MMSC receives a message addressed to a recipient whose phone number length is less than or equal to ShortcodeMaxLength, no international conversion will be applied to the phone number before routing. The default value is 4.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSAccountingURL=http://server/path

Before the gateway accepts an SMS or MMS message for delivery, the gateway will connect to a configurable customer-provided URL, providing information about the message to be sent, and who is sending the message. The customer provided URL can either tell the gateway to either accept or reject the message. This is a "pre-authorisation" request, and does not mean that the message will actually be accepted for delivery. If the gateway cannot successfully connect to the URL, or the URL returns a response other than a standard "HTTP 200 OK", the user request to send a message will be blocked. A "PreAuth" request to send a message will also be blocked if the response includes the text "PreAuth=Deny".

After an SMS or MMS message is accepted for delivery, the gateway will connect to a configurable customer provided URL, providing accounting information about the message being sent, allowing the customer to maintain external accounting information on messages processed by the Now SMS/MMS Gateway. The gateway ignores HTTP responses to the accounting callbacks.

See http://www.nowsms.com/support/bulletins/tb-nowsms-001.htm

MMSRoutingURL=http://server/path

This setting enables an HTTP-based routing callback for dynamic routing of MMS messages. When the MMSC receives a message, it will connect to a configurable customer-provided URL, passing the message recipient to the URL. The customer provided URL can return a response to indicate that the message should be routed via a specific route defined in the "MMSC Routing" page of the NowSMS configuration dialog.

The variables listed below will be added to the MMSRoutingURL when the URL is executed by the gateway as HTTP GET (CGI-style) parameters.

Type=MMSRouteCheck (Note: Future "Type" values may be added in the future.)
From=SenderPhoneNumber or e-mail address
VASPIN=VASPname (present if the message was received via a specific account defined in the "MMSC VASP" list)
To=RecipientPhoneNumber

Example:

```
http://server/path?Type=MMSRouteCheck&From=%2B1234567&VASPIN=test&To=%2B99999999
```

(Note: The "%2B" in the above examples is URL escaping for the "+" character.)
To specify which of the routes defined in the "MMSC Routing" list should be used to route this message, the URL must return a standard HTTP 200 OK response, and include the following text somewhere in the response:

Route=xxxxxxx

"xxxxxxx" should match an "Account Name" defined in the "MMSC Routing" list, or it can use the predefined values of "Direct" (signifying MMSC Direct Delivery) or "WAPPush" (signifying "Convert to Multimedia WAP Push").

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If it is possible to route the message via one of several defined routes, multiple routes can be returned in the response, separated by ":". For example:

Route=xxxxxxx:yyyyyy:zzzzzz

DisableMMSUserStats=Yes

(v2006+) This setting disables the MMSC from maintaining MMSC user statistical information (*.QTA files under the MMSCUSERS directory structure).

DefaultMMSVersion=1.0 (or 1.1 or 1.2 or 1.3)

(v2006+) This setting specifies the default MMS version to be assumed when sending an MMS message to a user account that is not defined to the MMSC. The default setting is 1.0. (Note: Support for 1.3 added in v2007.06.27)

PhoneNumberMaxLength=##

(v2006+) The MMSC will reject any MM1 recipients where the phone number of the recipient is longer than this value. The default is 20. (This is to prevent problems with some SMSCs that will choke on attempts to send SMS messages to phone numbers longer than 20 digits.)

DebugMaxSize=####

(v2006+) This setting specifies the maximum size in MB of the MMSCDEBUG.LOG, when it the debug log is enabled.

NonDeliveryNotificationTimeout=####

(v2007.06.27+) To enable support for MMS non-delivery notifications to be generated when an MMS message expires without being delivered/retrieved by the recipient, use this setting. #### is the number of minutes after which to consider a message as being expired.

SMTPExtraHeader=xxxx

(v2007.06.27+) Allows an extra header to be added to MMS messages that are routed to an e-mail address. (For example, "SMTPExtraHeader=X-Mms: True") To define extra headers that span multiple lines, use \r\n in the value to denote where a line break should occur.

AutoBCC=xxxxxx

(v2007.06.27+) This parameter will force all submitted MMS messages to be automatically BCC'd to an additional recipient. Usually this recipient would be a special short code routed to a VASP or to an e-mail address, where another application is processing the messages for archival, tracking, and/or regulatory requirements. To add an automatic BCC recipient, edit MMSC.INI, and under the [MMSC] header, add AutoBCC=xxxxxxxxxxxx, where xxxxxxxxxxxx is the address that should be added as an additional BCC recipient.

SMSEMailTemplate=@@FromAddress@@ /@@Subject@@ /@@Text@@

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

(v2007.07.23+) This parameter defines a template setting to allow some minor changes to the standard SMS to e-mail format (From /Subject /Message Text) used by NowSMS. To change the SMS to e-mail format, edit MMSC.INI, and add an SMSEMailTemplate= setting to define a template for the SMS to e-mail format. The following replacement variables are supported in the template: @@FromAddress@@ (e-mail address of sender), @@FromName@@ (full name of sender), @@Subject@@ (subject of message), @@Text@@ (text of message). The default template is SMSEMailTemplate=@@FromAddress@@ /@@Subject@@ /@@Text@@

MSISDNRecipientPrefixConvert=aaaa:bbbb,cccc:ddd

(v2007.11.13+) This configuration parameter supports MMS recipient address conversions based upon the prefix of the recipient phone number. This conversion is applied after the MMSC adds a country code to any recipient phone numbers. To add any conversions, edit MMSC.INI, and under the [MMSC] header, add MSISDNRecipientPrefixConvert=+1234:+2345,+6789:+987 If the MMSC encounters a recipient address that starts with "+1234", it will replace this part of the recipient address with "+2345" (e.g., +123456789 becomes +234556789).

AccountingKeepAlive=No

(v2009.02.16+) For improved performance, keep-alive sockets are enabled by default for MMS accounting callbacks (MMSAccountingURL). This setting is used to disable the use of keep-alive sockets for MMS Accounting callbacks if there is a problem with keep-alive sockets in a particular environment.

RoutingKeepAlive=No

(v2009.02.16+) For improved performance, keep-alive sockets are enabled by default for MMS routing callbacks (MMSRoutingURL). This setting is used to disable the use of keep-alive sockets for MMS Routing callbacks if there is a problem with keep-alive sockets in a particular environment.

SuppressMMSInDeliveryReport=Yes

(v2009.06.30+) When NowSMS receives an MMS message from an "MMSC VASP" account with a delivery report requested, and the "MMSC VASP" account is configured to delivery the message to the "MMS-IN" directory, NowSMS will automatically generate a delivery report back to the sender unless this parameter is configured.

ForceRoutingCallback=Yes

(v2009.06.30+) This configuration option to forces the MMS Routing callback ("MMSRoutingURL=" setting) to always be checked, even for numbers that are in the "MMSC Users" list. This ensures that in mobile number portability environments, any numbers that have been ported out to another carrier will be checked, even if the user was previously provisioned on the MMSC. If this setting is not enabled, NowSMS will only call the MMS Routing callback for numbers that are not in the "MMSC Users " list.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMSC.INI, [ShortCode] section:

Short codes can be used to map a short phone number to a longer phone number or e-mail address. Short codes can also be used to map an e-mail address to a short code that is route to a VASP.

For example, a short code of 200 could be rerouted to an e-mail address. To define short codes, create a section titled [ShortCode] in the MMSC.INI file. Under that section, create entries in the following format:

###=+44335830375474/TYPE=PLMN
###=user@domain.com

In the examples above, "###" is a numeric short code address, and the value after the "=" sign is the address the short code should be rerouted to, specified as either a phone number in MMS addressing format (/TYPE=PLMN) or as an e-mail address.

Similarly, to route an e-mail address to a short code that is routed via a VASP, use the following format:

user@domain.com=6000/TYPE=PLMN

Note: For routing short codes to a VASP, it is recommended that you define an "MMSC Outbound Routing" definition to connect to the VASP via an MMS protocol, such as MM7. Then define the short codes for the VASP in the "Route messages to this account for recipient phone number(s)" field within the route definition.

MMSC.INI, [IPNotify] section:

This section allows MMS notifications for select phone numbers to be routed via IP instead of via SMS. Under this section header, specify PhoneNumber=IP.Address (e.g., +9999999=192.168.1.200), to have MMS notifications for the specified phone number get routed to the specified IP address. (This feature is intended primarily for testing environments.)

Effective with v2007.06.27 and later, this setting is also supported for all other WAP Push related functionality of NowSMS, including OTA settings.

MMSC.INI, [HostNameSenderOverride] section:

Some mobile operators, such as Vodafone UK, require that content providers use different URLs for adult vs. non-adult content. When sending MMS or Multimedia WAP push via NowSMS, the URL that NowSMS generates includes the "Local Host Name or IP Address" that is specified on the "MMSC" page of the NowSMS configuration. It is now possible to override that host name in generated URLs based upon the MMS Sender address ("MMSFROM" parameter when using NowSMS proprietary URL submission) that is specified in a submitted message. Within the MMSC.INI, it is possible to define a section titled [HostNameSenderOverride]. Under that section, specify a sender address and the alternate

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

host name to use for notifications generated by that sender (e.g., +447778001212=212.100.225.164 or nowsms@now.co.uk=contentserver1.now.co.uk).

MMSC.INI, [TranslateText] section:

This section allows you to define translation text for the text prompts that are used in the WML that is generated for multimedia WAP Push.

Under this section, the following translations can be defined:

Multimedia Message=
Subject=
From=
Download image=
Download audio=
Download video=
Download file=

The text after the "=" sign will be used to replace the text before the "=" sign.

MMSC Outbound Routing Parameters

Parameters applying to an "MMSC Routing" definition are described below. The configuration files for an MMSC Outbound Route are located in the VASPOUT\accountname subdirectory of the NowSMS directory.

VASP.INI, [VASP] section:

MaxRecips=###

When a message is routed to an external MMSC connection, NowSMS will automatically break messages sent to multiple recipients into separate message instances for grouped recipients. The number of recipients per message instance is variable based upon the MMSC connection type. For MM1 connections, the default value is 5 recipients per message instance. For EAIF connections, the default value is 10 recipients per message instance. For MM7 connections, the value is 100 recipients per message instance. Different MMS providers may have different limits, and it may be necessary to modify the default settings. To modify this setting, it is necessary to manually edit the VASPOUT\accountname\VASP.INI file associated with the "MMSC Routing" definition. Under the [VASP] header, add MaxRecips=#### to specfy the number of recipients allowed per single MMS message instance.

UseKeepAlive=No
KeepAliveTimeout=###

(v2008.04.10+) (MM7, MM4, EAIF only) By default, NowSMS will use keep-alive connections for external MMSC connections, when possible. By default, if there are no further

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

messages to transmit over the connection, NowSMS will keep the connection open for approximately 90 seconds waiting for additional messages to transmit before closing the connection. If this support causes problems for a particular MMSC connection, keep-alive support can be disabled for a specific "MMSC Routing" connection by editing the VASPOUT\routename\VASP.INI file, and adding UseKeepAlive=No under the [VASP] header. Similarly, to change the keep-alive timeout, edit VASPOUT\routename\VASP.INI, and add KeepAliveTimeout=### to specify the number of seconds for which the keep-alive connection should be kept open.

SendLimit=x/y

(v2008.10.22+) SMSCSendLimit=x/y setting can specify that the gateway will send no more than x messages per y seconds. If y is not specified, then the default is 1. For example, to limit a connection to 1 message every 2 seconds, specify SendLimit=1/2. To limit a connection to 3 messages per second, specify SendLimit=3 or SendLimit=3/1.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Technical Bulletins

This section contains technical bulletins that provide supplemental information to the standard Now SMS/MMS Gateway documentation.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Now SMS/MMS Accounting Callbacks

Document ID: TB-NOWSMS-001, Last Update: February 5, 2003

Accounting callbacks have been added to the Now SMS/MMS Gateway effective with v4.11 and later.

Before the gateway accepts an SMS or MMS message for delivery, the gateway will connect to a configurable customer-provided URL, providing information about the message to be sent, and who is sending the message. The customer provided URL can either tell the gateway to either accept or reject the message. This is a "pre-authorisation" request, and does not mean that the message will actually be accepted for delivery. If the gateway cannot successfully connect to the URL, or the URL returns a response other than a standard "HTTP 200 OK", the user request to send a message will be blocked. A "PreAuth" request to send a message will also be blocked if the response includes the text "PreAuth=Deny".

After an SMS or MMS message is accepted for delivery, the gateway will connect to a configurable customer provided URL, providing accounting information about the message being sent, allowing the customer to maintain external accounting information on messages processed by the Now SMS/MMS Gateway. The gateway ignores HTTP responses to the accounting callbacks.

MMS Accounting Callbacks

For MMS messages, the MMS callback URL is defined in the MMSC.INI file, under the [MMSC] section header:

MMSAccountingURL=http://server.name/path

The variables listed in the sections below will be added to the MMSAccountingURL when the URL is executed by the gateway as HTTP GET (CGI-style) parameters.

MMSSend PreAuth Callback

This callback is executed when an MMS user is requesting to send a message. The following variables will be set:

PreAuth=Yes (indicates that the message is a Pre-Authorisation Request)
Type=MMSSend
From=SenderPhoneNumber
MsgCount=#### (number of recipients user is requesting to send the message to)

MMSSend Accounting Callback

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

This callback is executed after an MMS message that has been sent by an MMS user has been accepted by the gateway for delivery.

Type=MMSSend
From=SenderPhoneNumber
To=RecipientPhoneNumberOrEMailAddress
MessageID=SystemSpecificMessageID
Size=##### (size of message in bytes, may vary slightly with header changes in-transit)

MMSEMail Pre-Auth Callback

This callback is executed when an e-mail message is being presented for delivery to an MMS recipient. The following variables will be set:

PreAuth=Yes
Type=MMSEMail
From=SenderPhoneNumberOrEMailAddress (almost always an e-mail address)
To=RecipientPhoneNumber
MsgCount=1 (always 1 in current implementations)

MMSEMail Accounting Callback

This callback is executed after an MMS message has been received via e-mail, and has been queued for delivery to an MMS recipient. The following variables will be set:

Type=MMSEMail
From=SenderPhoneNumberOrEMailAddress (almost always an e-mail address)
To=RecipientPhoneNumber
MessageID=SystemSpecificMessageID
Size=#### (size of message in bytes, may vary slightly with header changes in-transit)

MMSRetrieve Accounting Callback

This callback is executed with a client connects in to retrieve an MMS message. The following variables will be set:

Type=MMSRetrieve
From=SenderPhoneNumberOrEMailAddress
To=RecipientPhoneNumber
MessageID=SystemSpecificMessageID
Size=#### (size of message in bytes)

SMS Accounting Callbacks

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

For SMS messages, the SMS callback URL is defined in the SMSGW.INI file, under the [SMSGW] section header:

SMSAccountingURL=http://server.name/path

The variables listed in the sections below will be added to the SMSAccountingURL when the URL is executed by the gateway as HTTP GET (CGI-style) parameters.

SMSSend PreAuth Callback

This callback is executed when an SMS (web, SMPP, SMTP) user is requesting to send a message. The following variables will be set:

PreAuth=Yes (indicates that the message is a Pre-Authorisation Request)
Type=SMSSend
From=DefinedSMSUserAccount
MsgCount=#### (number of recipients user is requesting to send the message to)

SMSSend Accounting Callback

This callback is executed after an SMS message that has been sent by an SMS user has been accepted by the gateway for delivery.

Type=SMSSend
From=DefinedSMSUserAccount
To=RecipientPhoneNumber
MessageID=SystemSpecificMessageID
Size=##### (size of message in bytes, may vary slightly with header changes in-transit)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Now MMSC Operator Considerations

Document ID: TB-NOWSMS-002, Last Update: May 5, 2003

Effective with v4.11 and later of the Now SMS/MMS Gateway, the operator functions of the Now MMSC are now included as base features of the MMSC built into the Now SMS/MMS Gateway.

The primary considerations for the Now MMSC in an operator environment pertain to user provisioning and user identification.

In the standard configuration for the MMSC, it is a requirement that each user be configured with a special MMS Message Server URL. The URL contains the username and password of the user account on the MMSC. In an operator environment, it is not practical to provision phones with unique MMS Message Server URLs, and it is also not practical to manually configure user accounts on the MMSC.

The reason that the standard configuration of the MMSC requires each user to be configured with a special MMS Message Server URL, is because the connection that the phone makes to the MMSC is IP-based, and contains no information about the MSISDN (phone number) of the user that is making the request. In fact, the MMSC only sees an HTTP request coming from the IP address of the WAP gateway that is proxying the request on behalf of the mobile device. Without further hooks into the operator network to determine the identity of the device making the request, the MMSC relies on the username and password in the URL request to identify the user.

To overcome this limitation, it is possible to configure the MMSC to receive user identity information directly from the operator network, without requiring the username and password on the URL.

To integrate into the operator network, the Now MMSC expects to receive user information from the operator WAP gateway. As all requests to the MMSC are being proxied through the operator WAP gateway, it is best that the WAP gateway take responsibility for user identification. Additional information on configuring the NowWAP Proxy to provide this information is provided later in this document.

The WAP gateway must be configured to forward the MSISDN of the requesting user to the MMSC via a configurable HTTP header. For example, the NowWAP Proxy uses the "X-MSISDN:" header to forward the MSISDN to HTTP content servers such as the MMSC.

To configure the Now MMSC to read the MSISDN from an HTTP header, configuration settings must be applied to the MMSC.INI file. The following configuration settings may be applied within the [MMSC] section of the MMSC.INI file:

MSISDNHeader=header-name

> The MSISDNHeader setting specifies the name of the HTTP header that will contain the MSISDN. For example, with NowWAP, the appropriate setting would be:
>
> MSISDNHeader=X-MSISDN

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MSISDNHeaderGateways=1.2.3.4,5.6.7.8,9.10.11.*

The MSISDNHeaderGateways setting specifies a list of one or more IP addresses from which the MMSC will accept the MSISDNHeader. This is to prevent forged requests, where another gateway or application inserts an MSISDN header to attempt to fool the MMSC. One or more IP addresses can be listed in this configuration setting. Each address must be separated by a comma. Wildcard addresses are supported by placing a "*" in place of a position within the IP address. For example, a setting of 9.10.11.* would mean that the MSISDNHeader would be accepted from any request originating from an IP address in the range of 9.10.11.1 thru 9.10.11.255.

MSISDNHeaderDefaultCountryCode=##

The MSISDNHeaderDefaultCountryCode setting specifies a default country code to be applied to MSISDN numbers presented in the MSISDNHeader, so that the gateway can convert the MSISDN address to international format automatically.

MSISDNHeaderLocalPrefix=#

The MSISDNHeaderLocalPrefix setting specifies the default prefix that is used for phone numbers in the MSISDN header that are in local (national) format. For example, in the UK, the MSISDNHeaderDefaultCountryCode would be 44, and the MSISDNHeaderLocalPrefix would be 0. With these settings, and MSISDN header of "07778001210" would automatically be converted to "+447778001210" by the MMSC.

MSISDNHeaderAutoProvision=Yes/No

The MSISDNHeaderAutoProvision setting specifies whether or not user accounts should be automatically provisioned on the MMSC the first time that a user sends a message through the MMSC. This removes the requirement to automatically provision accounts. Any user that makes a request through the appropriate WAP gateway with the MSISDNHeader set will be automatically provisioned on the MMSC.

Configuring the NowWAP Proxy to Forward MSISDN

When a mobile device connects to a WAP Gateway, the connection requests are made over the IP protocol, and there is no part of the WAP standard that specifies how the MSISDN is presented to the WAP gateway.

The mobile device is generally assigned an IP address when it connects to the GGSN (or network access server for dial-up connections, collectively we'll refer to them as this point on as an "access server"). This IP address is usually dynamically assigned and changes between sessions.

The WAP gateway needs to receive information from the "access server" in order to identify the MSISDN of a device associated with the IP address making a request of the WAP gateway.

The NowWAP proxy integrates with access servers using the Radius accounting protocol, which is one of the Radius (Remote Authentication Dial In User Service) protocols. The

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Radius authentication protocol is defined in internet RFC 2865, and the Radius accounting protocol is defined in internet RFC 2866. The two protocols are frequently used together.

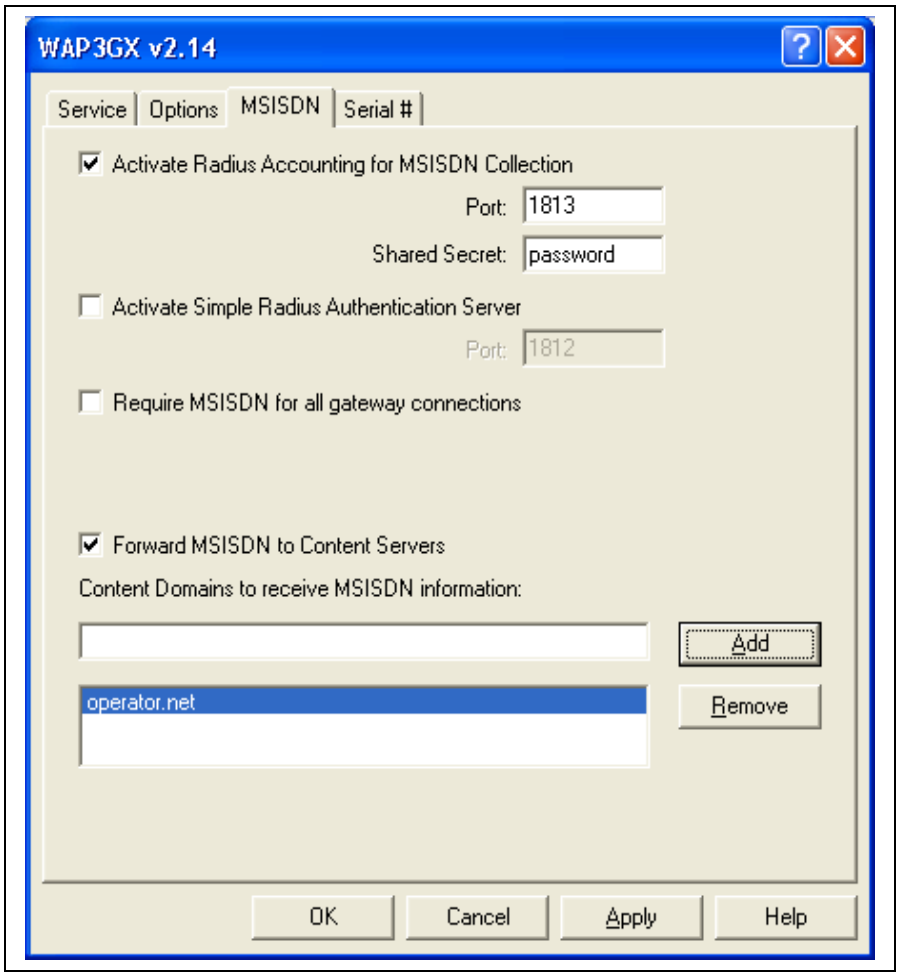
The Radius authentication protocol is used to authenticate a user. When a new connection is received by an access server (including a GGSN), the access server can be configured to use Radius authentication to determine whether or not to accept the connection. The access server sends a Radius authentication request over UDP to a Radius server. This request includes the username presented and a hash of the password against a shared secret, and usually the CLI (caller line identification) of the station making the request. The Radius server responds back over UDP to the access server telling the access server whether or not to allow the connection, and in some configurations, the Radius server can also tell the access server what IP address to assign to the client.

After a connection has been authenticated, the access server can be configured to use the Radius accounting protocol to inform an accounting server of a new connection. Similarly, the access server notifies of a dropped connection. The Radius accounting protocol is also UDP-based, but it is an informational protocol. After a new connection is authenticated, the access server sends a Radius accounting packet over UDP to the configured accounting server(s). This packet includes a hash of a shared secret for packet validation, and typically includes the username that connected, the IP address that was assigned, and the CLI of the station that connected.

In an operator environment, users are typically not assigned their own username for the purposes of Radius authentication. Instead CLI is the determining factor.

To provide MSISDN support, NowWAP needs to be configured as a Radius accounting server, so that it
receives Radius accounting packets from the access server. This support is activated in NowWAP on the MSISDN configuration dialog.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved



In the configuration dialog, you must specify a port (1813 is the default according to the specification, but some access servers default to 1646 which is an incorrect default from an older specification), and the appropriate shared secret.

This setting enables the NowWAP Proxy to receive Radius accounting packets from the access server.

Check "Forward MSISDN to Content Servers", and add your local domain name as a content domain. The
gateway will then include the MSISDN in any requests to content servers within any listed domains. For example, if "operator.net" is listed as a supported content domain, then the MSISDN would be forwarded to a content server named "mms.operator.net", or a content server simply named "operator.net". The gateway will forward the MSISDN to the content server in all HTTP requests sent to the content server, using the HTTP "X-MSISDN:" header.

As long as the Radius accounting feed from the access server is reliable, it may be desirable to require an MSISDN for any connections to the WAP gateway. This can be enabled by checking "Require MSISDN for all gateway connections". Connections from IP addresses where the WAP gateway has not received an MSISDN assignment from the access server will be rejected.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Getting the Radius accounting feed setup in the access server correctly, so that it is sent to the WAP gateway is the hardest part of this process. Unfortunately, NowWAP doesn't give you much indication as to whether or not it is receiving a Radius accounting feed. When testing, it is best to enable the debug log in NowWAP (edit WAPGW.INI, add Debug=Yes under [WAPGW] section header), and monitor the WAPDEBUG.LOG for a keyword of "Radius", where it will log all Radius packets that the gateway receives. If you have problems getting the Radius configuration working, then it is best to contact NowWAP technical support for further advice.

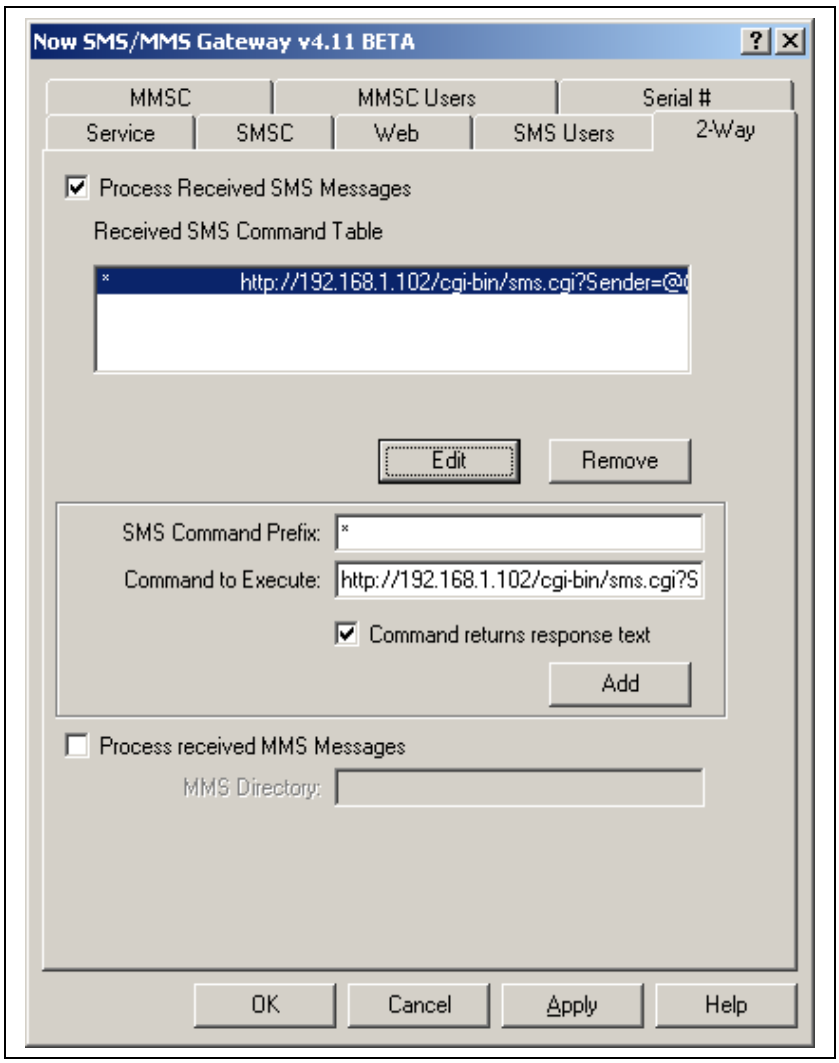
NowWAP can also be configured to be a Radius authentication server, but it is a VERY simple Radius authentication server. When NowWAP is a Radius authentication server, it will accept any authentication request. We do not recommend the use of this setting, except for testing purposes. (This setting was added originally for a particular customer that needed it.)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

2-Way SMS Returning a Non-Text Response

Document ID: TB-NOWSMS-003, Last Update: February 7, 2003

It is a relatively simple process to define a 2-way command that returns a text response to the message sender. The "2-Way" page of the NowSMS configuration dialog allows you to define commands that are executed when an SMS message is received. The receipt of an inbound SMS message by the gateway can cause an executable command to be executed, or the gateway can connect to an HTTP URL to run a script on a separate web server. If "Command returns response text" is set for the command, it is simple for the command to return a text response back to the sender of the SMS. In the case of an HTTP URL, the HTTP URL can return a simple text response with a MIME type for the response set as "text/plain", and the gateway will automatically send this text as a response back to the sender of the SMS.



In the example shown in the screen shot above, the receipt of a text SMS message causes the gateway to connect to the HTTP URL:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

http://192.168.1.102/cgi-bin/sms.cgi?Sender=@@Sender@@&Text=@@FullSMS@@

When a message is processed, the "@@Sender@@" value is replaced with the phone number of the sender of the received message, and @@FullSMS@@ contains the complete text of the received message.

What if an HTTP script wants to return a response back to the sender that is a non-text SMS format, such as WAP OTA settings, a Nokia smart messaging message, or even an MMS message?

The HTTP script could always include its own processing logic to spawn a command that issues a request back to the NowSMS gateway, but this can be difficult. An alternative approach is for the HTTP response to return a "redirect" response, where the redirect URL is a URL pointer back to the NowSMS gateway, for example http://127.0.0.1:8800/?PhoneNumber=%2b12125551212&... (If you want to be able to test your HTTP script by executing it from a web browser, you will need to substitute the real IP address of your NowSMS server for the 127.0.0.1 address.)

If your NowSMS server requires authentication, a username and password can be included in the redirect response using the URL format http://user:password@server.name/path.

Any URL that would work when sent directly to the NowSMS server can be returned as a "redirect" response. If a script will be returning a "redirect" response, it is important that "Command returns response text" is NOT checked for the command on the "2-Way" configuration dialog. This is because a redirect response simply instructs the HTTP client to retrieve a different URL. The NowSMS gateway will not return the actual response from that URL request to the client.

An HTTP redirect response is an HTTP 302 response, with the redirect URL included in the "Location:" header, for example:

HTTP/1.0 302 Redirect
Location: http://127.0.0.1:8800/?PhoneNumber=...
<blank line>

Most scripting languages have built-in support for returning a redirect response. For example, in an ASP script, Response.Redirect ("http://127.0.0.1:8800/?PhoneNumber=...) is used to send a redirect response. Other scripting languages have similar support.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Using Port 80 on a PC running IIS

Document ID: TB-NOWSMS-004, Last Update: February 7, 2003

As mentioned on the NowSMS documentation, the MMSC works best with operator WAP gateways when it is configured to use HTTP port 80. HTTP Port 80 is the standard port used by web servers. If an IIS web server is running on the same PC as the NowSMS gateway, it is likely that port 80 is already in use by the IIS web server.

If you can bind multiple IP addresses to the PC running IIS and the NowSMS gateway, it is possible for both IIS and NowSMS to use port 80, by assigning one IP address to IIS, and another IIS to the NowSMS gateway.

To assign a specific IP address to the NowSMS gateway, it is a simple matter of selecting the appropriate IP Address from the "IP Address" list on the MMSC configuration dialog. This setting defaults to "(all available)", meaning that the NowSMS gateway would bind to the specified ports on all IP addresses bound to the PC. Note that when you configure NowSMS to use a specific IP address, it will no longer be accessible on the local machine using the loop back address 127.0.0.1, you must always use the specific IP address, even when connecting from the local machine. Also note that the "IP Address" setting on the MMSC configuration dialog applies only for the IP address used by the HTTP and SMTP ports of the MMSC. The Now SMS gateway itself also has an HTTP web port. The IP address setting for that interface is found on the "Web" configuration dialog.

It is not as simple to configure IIS to bind to only a specific IP address. IIS has a feature called "socket pooling", which causes it to grab the web ports on all IP addresses configured on the machine, even if web services have only been configured for a specific IP address.

To configure IIS to bind only to a specific IP address or a limited number of specific IP addresses, check the configuration of each host defined in the Internet Information Services Manager, and ensure that the host is configured for a specific address instead of all unassigned addresses. You must then disable IIS socket pooling.

For information on disabling socket pooling on IIS, and configuring IIS to connect to only a specific IP address, please consult the following Microsoft references:

IIS Binds To All Available IP Addresses When It Starts
http://support.microsoft.com/support/kb/articles/Q259/3/49.ASP

How to Disable Socket Pooling
http://support.microsoft.com/support/kb/articles/Q238/1/31.ASP

XCON: How to Disable Internet Information Services Socket Pooling on the SMTP Service
http://support.microsoft.com/support/kb/articles/Q281/7/60.ASP

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Provisioning MMSC User Accounts via HTTP

Document ID: TB-NOWSMS-005, Last Update: February 7, 2003

With the v4.11 release of the Now SMS/MMS Gateway, an additional HTTP interface has been added for provisioning MMSC User Accounts, to compliment the existing menu driven "MMSC Admin" interface.

The interface is HTTP "GET" based, and requires that the HTTP request include an Authorization header. The Authorization header is set with the BASE64 encoded value of "username:password", where username and password are the values configured for the MMSC Admin interface.

HTTP /provision?
AdminAction=Action&PhoneNumber=number&Name=AliasName&Password=password&FullName=Full+Name HTTP/1.0
Authorization: username:password (BASE64 encoded)
<blank line>

Valid values for "AdminAction" are "Add", "Modify" and "Delete".

All actions require that a "PhoneNumber" be specified. (If you are configuring phone numbers in international format, be sure to remember that HTTP GET URL parameters must be URL encoded, so that "%2B" is used to represent the "+" character.) The other parameters (Name, Password and FullName) are optional.

Assuming that the user is authorized to issue a provisioning command (valid username/password, and supported IP address for originating the request), the server will return a normal HTTP response code of 200 with a MIME content type of "text/plain". The response will start with "OK" if the request was successful, or "ERROR" if the request failed.

Support for restricting access to the MMSC Admin and provisioning web functions by IP address was added in v4.11 of the Now SMS/MMS Gateway. Under the [MMSC] section header, use the AdminIPAddressList setting, which is a comma delimited list of IP addresses that are allowed access to the gateway. Do not include any spaces between addresses. To specify a wildcard address for a subnet, use the "*" character (for example, 192.168.1.* to allow all addresses in the 192.168.1 subnet).

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Faster GSM Modem Speeds with SMS over GPRS

Document ID: TB-NOWSMS-006, Last Update: February 10, 2003

With the v4.11 release of the Now SMS/MMS Gateway, the GSM modem interface has been optimised to support faster message transmission, particularly when used on devices and networks that support SMS over GPRS. With traditional SMS over GSM links, it was only possible to achieve an SMS transmission speed of 6 to 10 messages per minute, depending on network factors. However, with SMS over GPRS, and the v4.11 release of the Now SMS/MMS Gateway, we've tested GSM modem speeds of 30 messages per minute, or 1 message every 2 seconds, a speed that is 3 to 5 times faster than previously available with GSM modems.

While SMS over GPRS is still not nearly as fast as SMS speeds available with direct SMSC connections, SMS over GPRS does make the use of GSM/GPRS modems fast enough for a larger variety of applications.

To support SMS over GPRS, you require a mobile phone that can be configured to support SMS over GPRS, and you will need a subscription with a mobile operator that supports SMS over GPRS. (Many mobile operators are in the process of adding support for SMS over GPRS, as it can be more cost effective for them to implement on their networks.)

Our tests of 30 messages per minute over a GSM/GPRS modem were conducted with a SonyEricsson T68i connected to the NowSMS gateway with a serial cable. The mobile operator was T-Mobile in the USA. We configured the T68i to support SMS over GPRS by selecting Connect->Data. comm.->SMS access->GPRS.

It is anticipated that many more devices and modems will be introduced that support SMS over GPRS in the coming months. Below is a list of devices of which we are currently aware that support SMS over GPRS, with instructions for configuring the device to support SMS over GPRS. This should not be considered a comprehensive list of devices supporting SMS over GPRS. Consult the documentation for your device for additional information, and inquire with your network operator to determine if they support SMS over GPRS.

Ericsson T39/T65:

Settings->Data comm.->SMS access->GPRS

Ericsson T68m/SonyEricsson T68i:

Connect->Data comm.->SMS access->GPRS

Nokia 8310:

Menu->Services->Settings->Edit active service settings->GPRS Connection->Select "Always Online"

Menu->Messages->Message Settings->Default Profile->Use GPRS->Select "Yes"

Siemens S45/ME45:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Menu->Messages->Message Setup->SMS. Select Options. Select Change Settings. Select SMS via GPRS.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Routing MMS Notifications via a WAP Push Proxy Gateway

Document ID: TB-NOWSMS-007, Last Update: April 4, 2003

With releases v4.20 and later of the Now SMS/MMS Gateway, a configuration option has been added to allow MMS Notification messages to be routed via a WAP Push Proxy Gateway, instead of direct delivery over SMS. This configuration option can allow for improved integration of the MMSC with some existing WAP environments.

Note that the WAP Push Proxy Gateway (PPG) to which the MMSC connects must support the following characteristics:

1. If authentication is required (username/password), the PPG must support HTTP Basic Authentication

2. The ability to receive push data in 8-bit format, with support for the MIME type "application/vnd.wap.mms-message" (note that when this setting is enabled, the WAP Push option supported by the NowSMS web menu interface will also generate push messages via the configured PPG, and in these cases, the MIME types "application/vnd.wap.sic" and "application/vnd.wap.slc" will be used).

3. A push recipient address format of "WAPPUSH=PhoneNumber/TYPE=PLMN@domain", where "PhoneNumber" will contain the phone number of the push recipient, and "domain" is a configurable setting.

To configure the Now SMS/MMS Gateway to generate MMS notifications via a WAP Push Proxy Gateway, you must manually edit the MMSC.INI, and create a section of this file with the header "[WAPPPG]". The following settings are supported under the "[WAPPPG]" section header:

URL=http://host.name:port/path

This setting is required. It should point to a URL that is valid for submitting messages to the PPG. The host name may be entered as either a DNS host name or IP address, and the port setting defaults to "80" if not otherwise specified.

User=username

This setting is optional. It specifies a username for HTTP Basic authentication to be used to connect to the PPG.

Password=password

This setting is optional. It specifies a password for HTTP Basic authentication to be used to connect to the PPG. (Note: This setting is ignored if the "User" setting is not specified.

PushAddressDomain=domain

This setting is optional. It specifies the "domain" portion of the WAP Push recipient address (e.g., "WAPPUSH=PhoneNumber/TYPE=PLMN@domain"). If this setting is left blank

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

or not specified, the "@domain" portion will not be included in the WAP Push recipient address specification.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Sending SMS from the Command Line

Document ID: TB-NOWSMS-008, Last Update: July 13, 2004

From time to time we get asked for a command line interface for sending SMS messages via NowSMS.

A message thread on our discussion board provides a simple example that has been referenced frequently. The purpose of this document is to provide an updated script which offers more flexibility. This updated script allows recipient phone numbers, and the message text, to be specified on the command line.

Assuming that the script is saved as a file named sms.js, you would issue the following command:

```
cscript sms.js PhoneNumber1[,PhoneNumber2,...] SMS Message Text
```

(cscript.exe is the Windows Script Host, which is a component of Windows which should be located in the \Windows\System32 directory.)

Examples:

```
cscript sms.js +44777777777 This is a test message

cscript sms.js +44777777777,+44777777778 This is a test message to 2 recipients
```

The SMS.JS script file is displayed below. Save everything between the "---begin sms.js---" and "---end sms.js---" markers to a file named SMS.JS. Edit the NowSMSServerAddress, NowSMSUserName and NowSMSPassword variable settings as appropriate for your installation.

```
---begin sms.js---
/*
This is a command line script for sending an SMS message via NowSMS.

Below, you must substitute in the address of your NowSMS server, plus a valid
username and password for
an account defined in the "SMS Users" list of that server.

Note: This script uses the encodeURIComponent method introduced in Internet
Explorer 5.5. If you are running
Windows 2000 or an earlier version of Windows, you must have Internet Explorer
5.5 or later installed for this
script to work.
*/

var NowSMSServerAddress = "http://127.0.0.1:8800";
var NowSMSUserName = "test";
var NowSMSPassword = "test";

function HTTPGET(strURL)
{
var strResult;

try
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
{
// Create the WinHTTPRequest ActiveX Object.
var WinHttpReq = new ActiveXObject("Msxml2.XMLHTTP" /* or
"WinHttp.WinHttpRequest.5"*/);

// Create an HTTP request.
var temp = WinHttpReq.Open("GET", strURL, false);

// Send the HTTP request.
WinHttpReq.Send();

// Retrieve the response text.
strResult = WinHttpReq.ResponseText;
}
catch (objError)
{
strResult = objError + "\n"
strResult += "WinHTTP returned error: " +
(objError.number & 0xFFFF).toString() + "\n\n";
strResult += objError.description;
}

// Return the response text.
return strResult;
}

var strRequest;

if (WScript.Arguments.Count() < 2) {
WScript.Echo ("Usage: " + WScript.ScriptName + "
PhoneNumber1[,PhoneNumber2,...] Message Text\r\n");
WScript.Quit();
}

strRequest = NowSMSServerAddress + "/?PhoneNumber=" +
encodeURIComponent(WScript.Arguments(0)) + "&User=" +
encodeURIComponent(NowSMSUserName) + "&password=" +
encodeURIComponent(NowSMSPassword) + "&text=";

for (i=1; i<WScript.Arguments.Count(); i++)
{
if (i > 1) {
strRequest += "%20";
}
strRequest += encodeURIComponent(WScript.Arguments(i));
}


WScript.Echo(HTTPGET(strRequest));
---end sms.js---
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

WAP Push or OTA: Unknown Sender

Document ID: TB-NOWSMS-009, Last Update: August 20, 2004

Have you encounter a problem where your WAP Push or OTA configuration message displays "Unknown Sender"?

While this occurrence does not result in a loss of functionality, it does appear less that professional. *(If you have yet to experience this, send a WAP push message to a SonyEricsson P900, or some of the newer Nokia Series 60 devices.)*

NowSMS v5.51 now includes the ability to specify the initiator URI, which some phones will display as the sender of WAP push or OTA configuration messages.

Under the [SMSGW] section header of SMSGW.INI, the setting "WAPPushInitiatorURI=" can be used to set the "X-WAP-Initiator-URI:" header, and "WAPPushFlag=" can be used to the set the "Push-Flag:" header.

In most instances, you will want to set the WAPPushInitiatorURI value to a web URL, including the "http://" prefix, and if only a host name is specified in the URL, include a trailing backslash to avoid problems with some devices (e.g., use "http://www.nowsms.com/").

The WAPPushFlag can generally be omitted. When specified, values of 1, 2 or 3 are expected. According to the WAP Push protocol, a value of 1 indicates that the initiator URI has been authenticated by the Push Proxy Gateway. A value of 2 indicates that the content of the push is trusted by the Push Proxy Gateway. A value of 3 combines both of these settings.

"WAPPushInitiatorURI=" and "WAPPushFlag=" parameters can also be used in HTTP requests when submitting WAP Push or OMA OTA messages.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Using NowSMS as an MMSC in CDMA or CDMA2000 Environments

Document ID: TB-NOWSMS-010, Last Update: February 24, 2006

Delivering MMSC functionality in an CDMA or CDMA2000 environment can be challenging because the original WAP specifications for CDMA have technical requirements that require additional SMSC functionality.

(Technical note: CDMA2000 is different from WCDMA. CDMA2000 is an evolutionary upgrade path from CDMA. By contrast, WCDMA is an evolutionary upgrade from GSM/GPRS. CDMA2000 builds upon technical standards already deployed for CDMA, while WCDMA/UMTS builds upon technical standards already deployed for GSM/GPRS.)

MMS Message Notifications are generally sent via WAP Push over WDP (Wireless Datagram Protocol) over SMS. (It is also possible to deliver WAP Push over WDP over IP, but this is generally not practical unless there is an accessible database lookup for phone number to IP address translation. NowSMS currently can facilitate this IP only scenario, but in a way that is only suitable for test lab environments. This capability is described in a separate technical bulletin, and is outside of the scope of this bulletin.)

When a WAP Push message is larger than the network size limit for SMS (140 bytes in a GSM environment, and somewhat variable in CDMA environments), the WAP Push message must be segmented to enable it to be sent over multiple SMS messages.

In GSM environments, User Data Header (UDH) fields have been standardised to support the necessary segmentation, and the UDH parameters for segmentation are well understood and widely supported.

In CDMA environments, no standard for segmentation exists. Therefore, the WAP Forum defined its own standard for the sending of segmented WDP messages in a CDMA environment.

A key requirement of this standard is that each segment of the message must use the same CDMA SMS message id. However, the CDMA SMS message id field is a value that is generated by the SMSC, and cannot be set by an application that is submitting messages to the SMSC, such as an MMSC or WAP Push Proxy Gateway.

The authors of the WAP specification anticipated this limitation and defined a standard for WDP Adaptation over SMPP (WAP-159-WDPWCMPAdapt).

By default, NowSMS generates WAP Push messages (and MMS Notification messages which are based upon WAP Push) using a format that is specific to GSM (and WCDMA/UMTS) environments. However, there is also a configuration option defined on the "**Advanced Settings**" page of the SMPP configuration properties titled "**Use WDP Adaptation for WAP Push and MMS Notifications (required for CDMA)".** This setting is primarily used when connecting to a CDMA based SMSC. When this configuration option is enabled, NowSMS uses a protocol independent format known as "WDP Adaptation". This is usually the only practical option for delivering WAP Push messages in a CDMA environment. In this case,

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

NowSMS will submit messages to the SMPP server using the "WAP" teleservice (SMPP service type), consistent with the WDP and WCMP Adaptation specification.

By default, even if a WAP push message is longer than 140 bytes, when this option is selected, NowSMS will deliver the complete message in a single submission to the SMSC. NowSMS then expects the SMSC to perform any necessary segmentation for delivering the message over SMS. (This expectation exists because the SMSC must use the same CDMA SMS message id for each segment of the message, and there is no place in the SMPP protocol for NowSMS to specify this message id.)

It is also possible to configure NowSMS to use segmentation for longer messages, even if WDP Adaptation is enabled by checking the "**Use TLV parameters for port numbers and segmentation**" option. In this case, NowSMS segments on a 140 byte boundary, the same as in GSM environments.

When configuring NowSMS in CDMA environments, segmentation issues are usually the biggest problem, because of the WAP protocol requirement that the SMSC use the same CDMA SMS message id for each segment of the message.

If you experience problems sending WAP Push and/or MMS notifications in a CDMA environment with WDP Adaptation, here are some suggestions:

Verify that the SMSC supports WDP Adaptation (WAP-159-WDPWCMPAdapt). If you are connecting via an independent SMS provider, it is likely that they are not aware of this protocol option, but it is likely to be supported by the CDMA operator's SMSC.

Instead of troubleshooting by sending MMS messages, try sending simple WAP push messages first. Send a simple WAP push message via the NowSMS web interface (page 90), keeping the URL and text relatively short to ensure that the message can be delivered in a single segment. If that works, next try increasing the amount of text in the message. If the message delivery fails with longer WAP push messages, then you will want to take steps to limit the size of MMS notification messages that are generated by the NowSMS MMSC. This can be accomplished with the following suggestions:

1.) Use the shortest possible host name (in number of characters) for the "Local Host Name or IP Address" setting of the MMSC. This host name must be included in the every MMS Notification that is generated by the MMSC. Every byte saved in the host name saves a byte in the size of the resulting notification messages.
2.) If possible, use Port 80 as the "HTTP Port" for the MMSC. If a port other than port 80 is used, it must be appended to the host name when generating the URL.
3.) Edit MMSC.INI, and under the [MMSC] header, add CompactMMSURL=Yes. This setting shortens the length of the dynamic path that is generated when sending MMS messages and will save a few bytes in every MMS notification with no other effect.
4.) Edit MMSC.INI, and under the [MMSC] header, add MMSNotificationNoSubject=Yes. By default, NowSMS will include the message subject in the MMS Notification message. Having the subject present can help someone who has a mobile phone configured for manual message download to determine whether or not they want to download the message. However, the specification does not require the subject to be present in the notification, and omitting it can prevent long MMS notifications.
5.) Edit MMSC.INI, and under the [MMSC] header, add Nokia3510Compatible=No. This setting reduces the size of the MMS notification by approximately 30 bytes, at the

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

expense of compatibility with early model GSM MMS phones (specifically the original Nokia 3510 and Panasonic GD87). These phones did not understand the short form of MMS content type encoding in the MMS header, and required the long form. The MMS specifications do require that the handsets support the short form of this encoding, and technically these early model handsets were non-conformant. While GSM operators may want to maintain compatibility with these early model handsets, this should not be a consideration in CDMA environments.

NowSMS 2006 has also been extended to provide extensions to the HTTP SMSC interface to facilitate MMS and WAP Push in CDMA environments. This functionality is intended primarily for test lab environments (specifically with Agilent's CDMA test kit), but might also be useful for mobile operators whose SMSCs do not support WDP Adaptation.

To enable this support, define an HTTP SMSC connection in NowSMS as normal.

Next, manually edit the SMSGW.INI, and locate the [HTTP – server:port] section for this HTTP SMSC connection. Under this header, add CDMATemplateWAP=, where this is a URL template to be used for submitting WAP Push messages (MMS notifications are a special type of WAP Push). This URL template is similar to the existing text and binary templates, however you should use the @@CDMAWAPPDU@@ replacement parameter for NowSMS to insert the WAP PDU in CDMA format. (Do not use the @@TEXT@@, @@DATA@@, @@DATABIN@@ or @@UDH@@ parameters!)

By default, NowSMS will segment the CDMA WAP PDU to a size limit of 112 bytes, and issue multiple HTTP requests to submit the message in multiple segments. To change this size limit, specify CDMASizeLimit=### in the same section of the SMSGW.INI file.

An additional replacement parameter, @@CDMAWAPMMTS@@ is available. This parameter will be set to "1" for all segments except the final segment in a multiple segment WAP Push submission. This parameter will be set to "0" in a single segment WAP Push submission, and also in the final segment of a multiple segment WAP Push submission. ("MMTS" stands for "more messages to send", and is used to indicate that an additional segment will follow. This is necessary because in CDMA, the SMSC must apply the same MESSAGE_ID value to all segments of a multiple segment message.)

If the HTTP SMSC does not support the MMTS flag, then you should follow the advice that was given earlier in this bulletin regarding how to limit the size of the MMS Notification message.

Additionally, ensure that the URL template sets whatever flag is necessary to indicate that the CDMA WAP teleservice should be used.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Delivering MMS Notifications with WAP Push over IP

Document ID: TB-NOWSMS-011, Last Update: February 24, 2006

The delivery of an MMS message to a mobile device is initially triggered by an MMS Notification message sent from the MMSC to the mobile device. This MMS Notification message contains header information about the MMS message, as well as an HTTP URL link (usually a dynamic link generated by the MMSC itself) that the device can use to retrieve the content of the MMS message.

The MMS Notification message is sent via the WAP Push protocol. The WAP Push protocol operates over the Wireless Session Protocol (WSP), which in turn is implemented over the Wireless Datagram Protocol (WDP).

WDP packets can be sent over any protocol for which WDP has been implemented, including IP and SMS.

In real world environments, for practical reasons, the WDP packets corresponding to an MMS Notification/WAP Push are delivered via SMS. The reason for this is quite simple, the MMSC does not know how to translate the phone number that should receive the notification into an IP address.

As described in another technical bulletin, the NowSMS MMSC can be configured to send its MMS Notifications via a separate WAP Push Proxy Gateway, and the WAP Push Proxy Gateway might be integrated into the network in such a way that it could decide whether to send the push via IP or SMS. That would be a decision for a separate WAP Push Proxy Gateway (PPG) component, rather than for the MMSC.

Still, we are frequently asked about the capability to deliver these WAP Push messages over IP instead of SMS. While we do not believe it is practical in a mobile environment at this time, it can offer a practical solution in test lab environments.

In NowSMS v5.50 and later, you can configure NowSMS to route MMS notifications for specific phone numbers to specific IP addresses. In the MMSC.INI file, it is possible to create an [IPNotify] section which contains a static phone number to IP address mapping table. Under this section header, specify PhoneNumber=IP.Address (e.g., +9999999=192.168.1.200) to have MMS notifications for the specified phone number get routed to the specified IP address.

Using this configuration,when an MMS message is to be delivered to a phone number that is defined in the [IPNotify] section of MMSC.INI, the MMS Notification message will be sent using connection-less WAP Push to UDP port 2948. As UDP does not offer guaranteed delivery, push messages will be periodically retried until an attempt is made to retrieve the message from the MMSC.

Beginning with the initial NowSMS 2006 release, MMS delivery notifications and read receipts will also be routed over this IPNotify interface, when appropriate. Both of these message types have a single delivery attempt, as there is no way to confirm delivery.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

NowSMS as a WAP Push Proxy Gateway

Document ID: TB-NOWSMS-012, Last Update: February 24, 2006

NowSMS v5.50 and later support the WAP Push Access Protocol (PAP) to provide the ability to generate more specialised types of WAP Push messages. PAP is the standard defined by the Open Mobile Alliance for how applications can interface to a Push Proxy Gateway for sending WAP Push messages.

With support for PAP, NowSMS can be used as a push proxy for other vendor's MMSCs. It also becomes possible to send other types of push messages, such as those related to SyncML, Wireless Village, and OMA Digital Rights Management.

An example of using PAP to send an OMA DRM rights message can be found in the following thread:

http://www.nowsms.com/discus/messages/485/3292.html

To post a push message via the PAP interface, perform an HTTP POST to the "web" port configured for the NowSMS web interface (not the MMSC), posting to a URL of "/pap".

A PAP post is a multipart MIME request, with the first MIME part being the PAP control document that specifies message recipient(s). The second MIME part is the data to be pushed.

An example of a service indication push being sent via PAP can be found below:

```
-----------------BEGIN EXAMPLE-----------------
POST /pap HTTP/1.0
Content-Type: multipart/related; boundary=mime-boundary; type="application/xml"
Content-Length: xxxxx

--mime-boundary
Content-Type: application/xml

<?xml version="1.0"?>
<!DOCTYPE pap PUBLIC "-//WAPFORUM//DTD PAP 2.0//EN"
"http://www.wapforum.org/DTD/pap_2.0.dtd"
[<?wap-pap-ver supported-versions="2.0,1.*"?>]>
<pap>
<push-message push-id="9fjeo39jf084@pi.com">
<address address-value="wappush=xxxxxxxxx/type=user@ppg.operator.com"></address>
</push-message>
</pap>
--mime-boundary
X-Wap-Application-Id: x-wap-application:wml.ua
Content-Type: text/vnd.wap.si

<?xml version="1.0"?>
<!DOCTYPE si PUBLIC "-//WAPFORUM//DTD SI 1.0//EN"
"http://www.wapforum.org/DTD/si.dtd">
<si>
<indication href="http://www.xyz.com/email/123/abc.wml" created="2004-06-25T15:23:15Z"
si-expires="2004-06-30T00:00:00Z">
You have 4 new emails
</indication>
</si>
--mime-boundary--
-----------------END EXAMPLE-----------------
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Note that the "/type=user@ppg.operator.com" included in the WAP Push address specification is optional for NowSMS, and will be ignored, as only the relevant destination phone number is parsed from the request.

NowSMS will perform XML to WBXML conversions for WAP push of the following content types:

text/vnd.wap.si
text/vnd.wap.sl
application/vnd.oma.drm.rights+xml
text/vnd.wap.emn+xml (or application/vnd.wap.emn+xml)
text/vnd.wap.co
text/vnd.wap.connectivity-xml

Other types (including "application/vnd.wap.mms-message") must be submitted via the appropriate binary encoding.

NowSMS supports the following text values for "X-Wap-Application-id":

x-wap-application:push.sia
x-wap-application:wml.ua
x-wap-application:wta.ua
x-wap-application:mms.ua
x-wap-application:push.syncml
x-wap-application:loc.ua
x-wap-application:syncml.dm
x-wap-application:drm.ua
x-wap-application:enm.ua
x-wap-application:wv.ua

Other "X-Wap-Application-id" header values must be submitted to NowSMS using their assigned decimal value rather than the text name.

Further information on the WAP Push Access Protocol can be found from by downloading the WAP-247-PAP specification from
http://www.openmobilealliance.org/tech/affiliates/wap/wapindex.html

Note that NowSMS does not support guaranteed delivery or delivery notifications.

NowSMS does not support base64 encoding for PAP content, binary content should be posted as 8-bit data using Content-transport-encoding: binary. We further recommend that the MIME part that includes the binary content have a "Content-Length:" header to avoid the potential of extra bytes (such as blank lines) being interpreted as part of the binary stream.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Provisioning SMS and MMSC User Accounts via HTTP

Document ID: TB-NOWSMS-013, Last Update: February 27, 2006

For NowSMS v5.51 and later, an HTTP-based interface has been added to allow for account provisioning by external applications. This interface is enabled when the web-based administration interface is enabled by checking "Enable Web Account Administration" on the "Web" page of the NowSMS configuration dialog.

This interface is accessed via the NowSMS web interface port, with a URI of "/provision" followed by a series of "HTTP GET" parameters. *(The admin username and password must be supplied in the HTTP GET request within an "Authorization:" header using HTTP Basic Authentication.)*

Two different provisioning request formats are defined, one for defining "SMS Users" accounts, and a second for defining "MMSC Users" accounts.

If a parameter of **"Type=SMS"** is specified, the following parameters are supported:

AdminAction=Add, Modify, Delete, or CreditCheck
Name=Account Name
Password=password
Fullname=Full Name
EnableWebLogin=Yes/No
EnableSmppLogin=Yes/No
EnableSmtpLogin=Yes/No
ForcedSenderAddress=Sender Address
RestrictIPAddress=list of IP addresses
EnableCreditBalance=Yes/No
CreditsToAdd=####
MessageLimitDefault=Yes/No
MessageLimitPerDay=####
MessageLimitPerMonth=####
RecipientAddresses=list of recipient phone numbers that should be routed to this account

As an example, to add credits to an existing account, issue the following URL request:

http://server:port/provision?Type=SMS&AdminAction=Modify&Name=accountname&CreditsToAdd=100

Assuming that the user is authorized to issue a provisioning command *(valid username/password, and supported IP address for originating the request)*, the server will return a normal HTTP response code of 200 with a MIME content type of "text/plain". The response will start with "OK" if the request was successful, or "ERROR" if the request failed. If a **"CreditsToAdd"** parameter was specified, or **"AdminAction=CreditCheck"**, the HTTP response will include the text **"Credits=####"**, where "####" is the current balance for the account.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If a parameter of "**Type=MMS**" is specified, the following parameters are supported:

AdminAction=Add, Modify, Delete, or CreditCheck
PhoneNumber=phonenumber
Name=Alias
Password=password
Fullname=Full Name
MessageLimitDefault=Yes/No
MessageLimitPerDay=####
MessageLimitPerMonth=####

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

MMS Virus Blocking

Document ID: TB-NOWSMS-014, Last Update: February 27, 2006

Various public reports have discussed the potential for viruses to be spread via MMS. To date, most of the identified viruses have targeted Nokia Series 60 (and potentially other Symbian) phones.

NowSMS v5.51b – March 2005

In March 2005, NowSMS v5.51b was released, which included a configuration option intended to help prevent the spread of viruses via MMS when NowSMS is used as the MMSC.

The text of the March 11, 2005 statement follows:

Public reports have identified a virus that can be spread via MMS on Nokia Series 60 (and potentially other Symbian) phones.

We have not received any confirmed reports of customers encountering this virus, however we take the current published reports seriously, and believe that there is a potential risk for additional variants of the current virus threat.

The current virus is known as "CommWarrior". It spreads as an infected Symbian application that is attached to an MMS message.

The recipient receives a message with a subject such as one of the following:

Norton AntiVirus Released now for mobile, install it!
Nokia ringtoner Nokia RingtoneManager for all models.
Security update #12 Significant security update. See www.symbian.com

The user is then presented with an option to "Install CommWarrior?". (And it is likely that the user will have to make an additional selection to confirm that they wish to install the application.)

If the user selects yes, then the CommWarrior application is installed on the Series 60/Symbian device. Of course, CommWarrior is actually a virus which after some delay, sends out infected MMS messages to other users in the individual's address book.

End users need to show discretion in installing any applications that they receive unsolicited. However, it is likely that some naive users will install the application allowing the virus to spread.

This virus specifically targets handsets that are using the Symbian OS, including Nokia Series 60 phones such as the 7610, 6600, 3650, 6260, and 7650. It cannot infect other types of handsets. And it can only infect a Symbian handset if the user elects to install the application that they received unsolicited in the MMS message.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If you have received an MMS message that prompted you to install an unknown application, especially CommWarrior, and you mistakenly installed the application on your phone, then you should take steps to remove the virus from your phone. For additional information on this current virus threat, and links to anti-virus vendors, see http://www.electricnews.net/news.html?code=9592732

For customers who are using NowSMS as an MMSC, we are presenting a security update to NowSMS which can block the delivery of executable attachments to MMS messages for any subscribers to the NowSMS MMSC ("MMSC Users"). Please note that this update is only relevant for configurations where NowSMS is being used as an MMSC in a somewhat public environment (such as an operator MMSC deployment).

Please see http://www.nowsms.com/discus/messages/53/8153.html for a complete list of changes in this update for the Now SMS/MMS Gateway v5.51. That thread also contains a link for the download.

The following text describes the new feature that is used to block the delivery of executable attachments to MMS messages.

MMSC: Add configuration option to block certain MIME types from being delivered to an MMS recipient when NowSMS is acting as the MMSC. This option is being implemented primarily to deal with potential MMS virus threats, where infected Symbian applications are being spread to Nokia Series 60 phones through MMS. To block executable MIME types, create a file named MMSBLOCK.TXT in the NowSMS program directory. In this file, list one MIME content type per line, specifying content types to be blocked. We recommend the following entries in this file to prevent Symbian and Java executables from being distributed via MMS:

- application/vnd.symbian.install
- application/java-archive
- application/x-java-archive
- text/vnd.sun.j2me.app.descriptor

This functionality requires that "Dynamic Image + Audio Conversion" be enabled for the MMSC.

NowSMS 2006 – March 2006

In the past year, since the release of the feature that could be used to help prevent the spread of MMS viruses when NowSMS is used as the MMSC, we have re-evaluated the technique that was originally introduced in NowSMS v5.51b (March 2005).

While the technique did help prevent the spread of a virus being distributed via MMS, it did not consider that most of these types of viruses are spread via multiple protocols. Many mobile viruses attempt to spread via Bluetooth, with MMS used as a secondary channel. When a phone becomes infected by a virus via Bluetooth, often the virus will then scan the contact list of the newly infected phone, and send attempt to send itself out to that phone's contact list via MMS. The NowSMS virus blocking efforts described above will prevent the virus from being spread. However, there are limitations of this solution:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

1.) The infected user still sends an MMS message out to all of their contacts. NowSMS only strips the executable component that contains the virus out of the message. While the virus is not spread, the infected user may still be charged for all of the MMS messages that were sent by the virus.

2.) The virus blocking feature relies on the "Dynamic Image + Audio Conversion" option being enabled in the MMSC, which is not desirable for all configurations.

3.) Trusted Value Added Service providers are unable to distribute any executable content via MMS.

Effective with the initial release of NowSMS 2006, the way in which the MMSBLOCK.TXT file is implemented has been changed to address these concerns.

Now, whenever an MMS message is submitted to the MMSC via MM1, MM4, or SMTP (MM3), NowSMS examines the message to determine if the message contains any content that is in the blocked list. For MM1 submissions, the MMSC will refuse to accept the message, so that the user will not be charged for the attempt. For MM4 and SMTP connections, the MMSC will process the message, but will skip any content that is in the blocked list.

The intent is to block the spread of viruses that are transmitted user to user. MM7 connections are allowed to send content of the content types that are defined in the MMSBLOCK.TXT file.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

NowSMS and OMA Digital Rights Management (DRM)

Document ID: TB-NOWSMS-015, Last Update: February 27, 2006

Note: This technical bulletin has been incorporated into the NowSMS 2007 Manual. Please see the **Digital Rights Management** section beginning on page 320.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

Receiving MMS Messages with a PHP Script: HTTP File Upload Post

Document ID: TB-NOWSMS-016, Last Update: November 3, 2006

NowSMS has long been a popular tool for enabling rapid development of interactive SMS applications and services. Within the NowSMS product, we refer to this as 2-way SMS. Through the 2-way SMS facility, when NowSMS receives an SMS message, it can be configured to dispatch that message to a script running on an HTTP server, to a local executable program, or local script or batch file. This provides a simple way to get received messages into an application, so that the application can perform custom processing on the message. The application can generate a simple reply back to the received message, or perform more advanced application specific logic. The scripts that process the received message can be written in popular web server scripting languages such as PHP, ASP and Perl, making the development of these scripts a relatively simple process for web developers.

NowSMS also supports the ability to route received MMS messages to an application. However, historically, it has been considerably more difficult to develop an application to support receiving MMS messages, as compared to SMS. The primary reason for this increased level of complexity is that the content of an MMS message is considerably more complex than that of an SMS text message, as a single MMS message can contain multiple content objects of different types.

NowSMS has historically supported three different techniques for delivering received MMS messages to an application:

- A File/Directory based interface where newly received MMS messages are placed in a directory on the NowSMS server. A header file contains a text version of the header of the MMS message, as well as pointers to separate files that contain the MMS message content (text, images, video, etc.).
- An XML/SOAP interface over HTTP POST where the MMS message content is packaged according to the MM7 format defined by the 3GPP.
- An e-mail based interface where the MMS message content is repackaged into an e-mail message format and routed to a specified e-mail address.

Receiving MMS via HTTP File Upload Post

A new technique for 2-way MMS has been added, beginning with the October 30, 2006 (2006.10.30) release of NowSMS 2006. This technique uses HTTP POST to transmit the received MMS message content to a script file running on another web server. The HTTP POST format that is used is the same format that is used for "HTTP File Upload" (multipart/form-data) from an HTML form, with particular considerations to make it easier to process the HTTP POST using the PHP scripting language. This technique is not specific to PHP, and we hope to provide examples for other scripting languages in the future.

When an MMS message is received, NowSMS can be configured to perform an HTTP File Upload Post to a configurable URL. The format of the HTTP File Upload POST is similar to the Now SMS/MMS Proprietary URL Submission Format that is used by the "Send MMS Message" form in the NowSMS web interface, as described at the following link:

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

http://www.nowsms.com/documentation/ProductDocumentation/mms_notifications_and_content/Submitting_MMS_Messages_URL.htm

There are two important differences that should be noted when comparing this HTTP File Upload Post to the Now SMS/MMS Proprietary URL Submission format:

- An **MMSText** variable is not present in the HTTP File Upload Post. Any text parts of the MMS message will be posted as file components with a MIME type of "text/plain".
- The **MMSFile** variable is replaced by **MMSFile[]** in the HTTP File Upload Post in order to allow the variable to be processed as an array by PHP scripts. (Effective with the 2006.10.30 release, NowSMS will also accept MMSFile[] as an alias for MMSFile when using the Proprietary URL Submission Format.)

The raw HTTP File Upload Post will look similar to the following:

```
POST /mmsreceive.php HTTP/1.0
Host: x.x.x.x
Content-type: multipart/form-data; boundary="--boundary-border--"
Content-length: xxxxx (size of content part of post, everything after HTTP header)
Connection: close
Authorization: username:password (base64 encoded)

----boundary-border--
Content-Disposition: form-data; name="PhoneNumber"

+44123456789

----boundary-border--
Content-Disposition: form-data; name="MMSFrom"

+44987654321
----boundary-border--
Content-Disposition: form-data; name="MMSSubject"

Message Subject
----boundary-border--
Content-Disposition: form-data; name="MMSFile[]"; filename="original-filename.ext"
Content-type: Mime/Type

File data goes here
----boundary-border--
Content-Disposition: form-data; name="MMSFile[]"; filename="original-filename2.ext"
Content-type: Mime/Type

The MMSFile[] part may be repeated for multiple objects within the MMS message.
----boundary-border----
```

The URL to which the data is posted (http://x.x.x.x/mmsreceive.php in the above example) is configurable.

The actual MIME "boundary=" value will vary.

The "Content-Length:" header refers to the size in bytes of the content part of the HTTP POST (that is, all content that follows the HTTP header).

The "Authorization:" header will only be present if a username/password are configured for the POST within the NowSMS configuration.

The "PhoneNumber" variable contains the message recipient phone number (usually the phone number associated with the GSM/GPRS modem if messages are being received over a modem interface).

The "MMSFrom" variable contains the message sender address.

The "MMSSubject" variable contains the subject of the MMS message.

The "MMSFile[]" variable contains the raw file content of individual objects within the MMS message, and may be repeated multiple times.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

PHP Processing of MMS HTTP File Upload Post (mmsreceive.php)

PHP provides built-in functionality for processing an HTTP File Upload Post. This functionality is described in the PHP manual in the chapter titled "Handling HTTP File Uploads", which is currently accessible on-line at http://php.net/manual/en/features.file-upload.php.

The global $_FILES array provides access to the file uploads, and the move_uploaded_file function allows you to save a copy of an uploaded file to another location. Pay particular attention to the example in the PHP manual that shows uploading an array of files, as this is the technique that is used for processing the multiple "MMSFILE" components of an MMS message.

The following is an example PHP script that processes received MMS messages and produces a simple HTML blog containing all images and other content submitted by remote users, with each user having their own blog. This PHP script (mmsreceive.php) can be downloaded as part of a ZIP file at http://www.nowsms.com/download/php2waymms.zip.

```
<?php

/* Replace the directory below with the directory that you wish to store received MMS messages in */
$upload_path = "c:\\upload\\";

/* By default, the script will return an HTTP 500 error if there are any internal processing errors,
    such as a problem creating a directory or file.  Returning an error in this way can signal the
submitting
    application that there is a problem and that it may need to retry.  However, it can make it
difficult
    to debug problems when testing submissions via a web form.  Set this variable to False to disable
the
    returning of an HTTP 500 error code.    */
$returnHttp500OnError = True;

/* Variables used by script */
$errorFlag = False;       /* Set to True if an error has occurred */
$dateString = date("YmdHis");  /* MMS images will be stored in a temporary directory name based upon
current date/time */
$savedImageFile = False;   /* If an error occurs, or the message is text only, delete the temporary
directory */

/* MMSFrom variable contains the sender phone number - REQUIRED */
if (!isset($_REQUEST['MMSFrom']) || !$_REQUEST['MMSFrom']) {
   echo "ERROR: MMSFrom variable (sender) is not set";
   $errorFlag = True;
}

/* MMSSubject variable contains the message subject - if not set, use default text */
if (!isset($_REQUEST['MMSSubject']) || !$_REQUEST['MMSSubject']) {
   $MMSSubject = "Multimedia Message";
}
else {
   $MMSSubject = $_REQUEST['MMSSubject'];
}


/* Validate that there is one or more HTTP file upload in the MMSFile[] array */
if (!$errorFlag) {
   $errorFlag = True; /* Default to returning an error, unless we find a valid HTTP file upload */
   if ($_FILES["MMSFile"] && is_array($_FILES["MMSFile"]) && count($_FILES["MMSFile"])) {
      foreach ($_FILES["MMSFile"]["error"] as $key => $error) {
         if ($error == UPLOAD_ERR_OK) {
            $errorFlag = False; /* reset error condition, found a valid HTTP file upload */
         }
      }
   }
   if ($errorFlag) {
      echo "ERROR:  Request does not include any uploaded files!";
   }
}

/* Build a directory for each user beneath the $upload_path, verify that we have rights to create a
temporary file in this user directory */
if (!$errorFlag) {
   $user_path = $upload_path . $_REQUEST['MMSFrom'];
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
   if (!@file_exists ($user_path)) {
      @mkdir ($user_path);
   }
   $user_path = $user_path . "\\";
   $tmp_filename = $user_path . "temp.tmp";
   $tmp_handle = @fopen ($tmp_filename, "w+");
   if (!$tmp_handle) {
      if ($returnHttp500OnError) header("HTTP/1.0 500 Internal Error");
      echo "ERROR:  Cannot create files in upload directory " . $user_path;
      $errorFlag = True;
   }
   else {
      fclose ($tmp_handle);
      @unlink ($tmp_filename);
   }
}

/* Build a temporary directory beneath the user directory for storing image files associated with this
MMS message */
if (!$errorFlag) {
   $image_path = $user_path . $dateString;
   if (!@file_exists ($image_path)) {
      @mkdir ($image_path);
   }
   $image_path = $image_path . "\\";
   $tmp_filename = $image_path . "temp.tmp";
   $tmp_handle = @fopen ($tmp_filename, "w+");
   if (!$tmp_handle) {
      if ($returnHttp500OnError) header("HTTP/1.0 500 Internal Error");
      echo "ERROR:  Cannot create files in upload directory " . $image_path;
      $errorFlag = True;
   }
   else {
      fclose ($tmp_handle);
      @unlink ($tmp_filename);
   }
}

if (!$errorFlag) {
   /* msglog.txt contains previous messages posted by this user in HTML format, without HTML headers.
      If this file exists, we copy it to msglog.tmp, so that we can create a new
      msglog.txt file with this new message at the top. */
   $msglogTxt = $user_path . "msglog.txt";
   $msglogTmp = $user_path . "msglog.tmp";
   if (@file_exists ($msglogTmp)) {
      @unlink ($msglogTmp);
   }
   if (@file_exists ($msglogTxt)) {
      if (!@copy ($msglogTxt, $msglogTmp)) {
         if ($returnHttp500OnError) header("HTTP/1.0 500 Internal Error");
         echo "ERROR:  Cannot create temporary file in upload directory " . $user_path;
         $errorFlag = True;
      }
   }
   if (!$errorFlag) {
      $msglogTxt_handle = @fopen ($msglogTxt, "w+");
      if (!$msglogTxt_handle) {
         if ($returnHttp500OnError) header("HTTP/1.0 500 Internal Error");
         echo "ERROR:  Cannot create temporary file in upload directory " . $user_path;
         $errorFlag = True;
      }
   }

   if (!$errorFlag) {
      /* This is where we take the MMS message content, and convert it to simple HTML */
      fwrite ($msglogTxt_handle, "<!-- Begin message block -->\r\n");
      fwrite ($msglogTxt_handle, "<h2>" . $MMSSubject . "</h2>\r\n");
      fwrite ($msglogTxt_handle, "<p><small>" . date("F j, Y, H:i") . "</small></p>\r\n");

      /* Repeat for each object/file within the MMS message */
      foreach ($_FILES["MMSFile"]["error"] as $key => $error) {
         if ($error == UPLOAD_ERR_OK) {
            $tmp_name = $_FILES["MMSFile"]["tmp_name"][$key];
            /* If the content is text, put it directly into the HTML */
            if (!strcmp (strtolower($_FILES["MMSFile"]["type"][$key]), "text/plain")) {
               fwrite ($msglogTxt_handle, "<p>" . file_get_contents ($tmp_name) . "</p>\r\n");
               echo "The file ".  basename( $_FILES["MMSFile"]["name"][$key]). " has been
uploaded<br/>";
            }
            /* If the content is SMIL, ignore it */
            else if (!strcmp (strtolower($_FILES["MMSFile"]["type"][$key]), "application/smil")) {
```


© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

```
               echo "The file ".  basename( $_FILES["MMSFile"]["name"][$key]). " has been
skipped<br/>";
            }
            else {
               /* If content is an image, reference it with an img tag, otherwise include an a href */
               $new_name = $image_path . basename( $_FILES["MMSFile"]["name"][$key]);
               if (@move_uploaded_file($tmp_name, $new_name)) {
                  $savedImageFile = True;
                  echo "The file ".  basename( $_FILES["MMSFile"]["name"][$key]). " has been
uploaded<br/>";
                  if (!strncmp (strtolower($_FILES["MMSFile"]["type"][$key]), "image/", 6)) {
                     fwrite ($msglogTxt_handle, "<p><img src=\"" . $dateString . "\\" .
                              basename($_FILES["MMSFile"]["name"][$key]) . "\"/></p>\r\n");
                  }
                  else if (!strncmp (strtolower($_FILES["MMSFile"]["type"][$key]), "video/", 6)) {
                     fwrite ($msglogTxt_handle, "<p>Video attachment:  <a href=\"" . $dateString .
"\\" .
                              basename($_FILES["MMSFile"]["name"][$key]) . "\">" .
                              basename($_FILES["MMSFile"]["name"][$key]) .
                              "</a></p>\r\n");
                  }
                  else {
                     fwrite ($msglogTxt_handle, "<p>File attachment:  <a href=\"" . $dateString . "\\"
.
                              basename($_FILES["MMSFile"]["name"][$key]) . "\">" .
                              basename($_FILES["MMSFile"]["name"][$key]) .
                              "</a></p>\r\n");
                  }
               }
               else {
                  echo "Error uploading file ".  basename( $_FILES["MMSFile"]["name"][$key]) .
"<br/>";
               }
            }
         }
      }
      fwrite ($msglogTxt_handle, "<!-- End message block -->\r\n");

      /* If we saved previous message entries to msglog.tmp, we need to add them back to this file */

      if (@file_exists ($msglogTmp)) {
         fwrite ($msglogTxt_handle, file_get_contents ($msglogTmp));
         @unlink ($msglogTmp);
      }
      fclose ($msglogTxt_handle);
   }

   /* If we didn't save any images (message is text only), then delete the temporary directory */
   if (!$savedImageFile && $image_path) {
      @rmdir ($user_path . $dateString);
   }

}

/* Now we put the HTML footers in place, creating index.html in the user directory */

if (!$errorFlag) {
   $htmlFile = $user_path . "index.html";
   $htmlFile_handle = @fopen ($htmlFile, "w+");
   if (!$htmlFile_handle) {
      if ($returnHttp500OnError) header("HTTP/1.0 500 Internal Error");
      echo "ERROR:  Cannot create file in user upload directory " . $user_path;
      $errorFlag = True;
   }
   else {
      fwrite ($htmlFile_handle, "<html>\r\n<head>\r\n<title>MMS Message Log for " .
$_REQUEST['MMSFrom'] .
                "</title>\r\n</head>\r\n<body>\r\n");
      fwrite ($htmlFile_handle, "<h1>MMS Message Log for " . $_REQUEST['MMSFrom'] . "</h1>\r\n");
      fwrite ($htmlFile_handle, file_get_contents ($user_path . "msglog.txt"));
      fwrite ($htmlFile_handle, "</body>\r\n</html>\r\n");
      fclose ($htmlFile_handle);
   }

}
```

The script begins by performing some parameter validation.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The script tests to ensure that the “MMSFrom” variable is present and non-blank, which includes the phone number of the message sender. If this variable is not present or blank, the script returns an error.

The script tests to see if the “MMSSubject” variable is present. If this variable is not present, or blank, a default subject of “Multimedia Message” is used.

The script tests the “MMSFile” array to determine if there are any uploaded files associated with the request. An error is returned if there are no uploaded files included in the request.

The script creates a directory for the user if one does not already exist. User directories are created beneath the directory specified by the $upload_path variable.

The script verifies that it has the ability to create files in the user directory. If there is an error creating the user directory, or creating a temporary file in the user directory, the script returns an error.

The script creates a subdirectory beneath the user directory to contain content associated with the current MMS message. A separate subdirectory is created for each message because over time it is possible that the user may send in multiple files with the same name, and for this example we want to preserve both the new and old content, while also preserving the original file name. Creating a separate directory for each received MMS message is a simple way of accomplishing these ends. The script verifies that it can create this directory and that it can create temporary files in the newly created directory.

Within the user directory, the script maintains two files: msglog.txt and index.html

index.html is an HTML file that contains a log of all previously received MMS messages, converted into an HTML format. Messages are listed in this file in an order of newest to oldest.

msglog.txt contains a log of all previously received MMS messages, similar to index.html. However, this file does not contain the complete HTML, it contains only the converted messages without all necessary HTML headers. The script maintains this file, in addition to index.html to simplify the processing required to regenerate index.html each time a new MMS message is received.

After the script has validated the input parameters, it creates a backup of the msglog.txt file named msglog.tmp.

It then creates a new msglog.txt file, looping through the content of the MMS message and reformatting the MMS message into a simple block of HTML. The block of HTML begins with the subject line of the MMS message as a header, followed by a date/time stamp. The script then loops through the content of the MMS message. Text files that are part of the MMS message content are included directly in the HTML content. Image files are included as a direct image reference within the HTML content. SMIL files are skipped and discarded. Other content is included via a link within the HTML content.

After completing the processing of the MMS message, the script appends the previously received messages that were backed up to the msglog.tmp file to the msglog.txt file. The script then inserts the necessary HTML headers to create index.html based upon the content of the msglog.txt file.

Notes about installing/configuring PHP and testing the PHP script

Use of the mmsreceive.php script assumes that you have some familiarity with PHP. You can find more information about PHP at http://www.php.net.

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

If you are installing PHP on a Windows platform, an installation guide at http://www.iis-resources.com/modules/AMS/article.php?storyid=615 can be quite useful. The only real flaw in that installation guide is that when it registers the ".PHP" extension with IIS, it assumes that PHP is the only type of dynamic content that will be served by your web server. To manually register the ".PHP" extension with IIS, right click on "Web Sites" in the IIS Manager, select "Properties", go to the "Home Directory" page, press the "Configuration" button, and here you can register the ".PHP" extension to use the "php5isapi.dll" executable.

Another potential headache in setting up PHP is that the web server needs to be configured to allow your script to write to the $upload_path directory in which you will place the received MMS message content. There are user contributed notes to the section of the PHP manual that documents "Handling HTTP File Uploads", which is currently accessible on-line at http://php.net/manual/en/features.file-upload.php, which are quite helpful in this regard. Essentially, you must give the IIS user account (IUSR_yourservername) read, write and directory browse access to your upload directory.

These issues are outside the scope of NowSMS. To help you troubleshoot your script, we recommend that while developing and testing your script, instead of testing it initially with live MMS content, you test with simulated MMS content. A simple HTML web form can be deployed to test performing an HTTP File Upload Post to your PHP script. We have included an example simple HTML web form (mmsreceive.html) in the ZIP file download that contains the PHP script described by this document (http://www.nowsms.com/download/php2waymms.zip).

This web form is shown below. As you can see, this is a very simple form that allows you to input values for the MMSFrom and MMSSubject variables, as well as up to 12 uploaded files to be passed to your script as the MMSFile[] array.

```
<html>
<head>
<title>File Upload Test for MMSReceive.php Script</title>
</head>
<body>
<form enctype="multipart/form-data" action="mmsreceive.php" method="POST">
<h1>File Upload Test for MMSReceive.php Script</h1>
<input type="hidden" name="MAX_FILE_SIZE" value="10000000" />
Sender Phone Number:  <input name="MMSFrom" type="text" /><br />
Message Subject:  <input name="MMSSubject" type="text" /><br />
Choose one or more files to upload:
<br/>
<input name="MMSFile[]" type="file" /><br />
<input name="MMSFile[]" type="file" /><br />
<input name="MMSFile[]" type="file" /><br />
<input name="MMSFile[]" type="file" /><br />
<input name="MMSFile[]" type="file" /><br />
<input name="MMSFile[]" type="file" /><br />
<input name="MMSFile[]" type="file" /><br />
<input name="MMSFile[]" type="file" /><br />
<input name="MMSFile[]" type="file" /><br />
<input name="MMSFile[]" type="file" /><br />
<input name="MMSFile[]" type="file" /><br />
<input name="MMSFile[]" type="file" /><br />
<input type="submit" value="Upload File" />
</form>
</body>
</html>
```

Configuring NowSMS to call your PHP Script

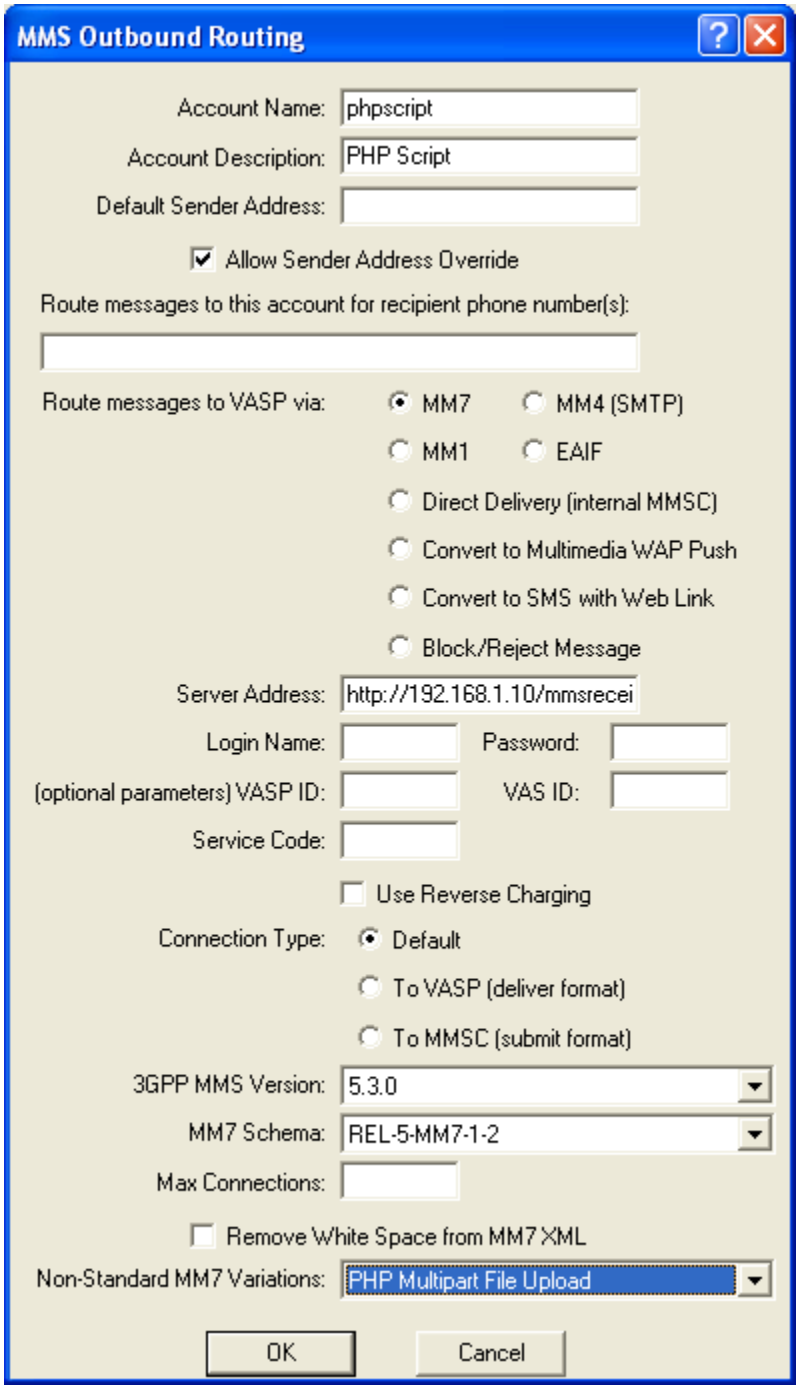
It should be noted that we left this important detail to be covered last. The reason for this decision is simple. You should thoroughly test your script before configuring NowSMS to connect to it. Even if you are only using the test script that we have provided in this example, you should test the script with the sample web form before configuring NowSMS

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

to connect to it. The reason we recommend this is because it is much easier to test and debug problems with the script, or with your PHP configuration in this manner. It is more difficult to test and debug problems when NowSMS is generating automated posts that are delivering actual MMS messages.

The first step of configuring NowSMS to connect to your PHP script is to define a connection to the script on the "MMSC Routing" page of the NowSMS configuration dialog. Press "Add" to add a routing.

The connection to a PHP script is defined as an external connection from the NowSMS MMSC, and it is defined similar to the process of defining an MM7 connection.



© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

The “Account Name” and “Account Description” parameters can contain any value (use letters and numbers only). These values are used only for identifying the connection to the PHP script within NowSMS.

“Default Sender Address” should be left blank, and “Allow Sender Address Override” should be checked.

“Route messages to this account for recipient phone number(s)” should be left blank in most configurations.

“Route messages to VASP via” should be set to MM7.

“Server Address” should contain the URL for your PHP script (e.g., http://server/mmsreceive.php).

“Login Name” and “Password” should be left blank unless your script or server requires user authentication. If these parameters are non-blank, NowSMS will use them to build a basic authentication “Authorization:” header to be used when connecting to your PHP script.

“VASP ID”, “VAS ID”, “Service Code”, “Use Reverse Charging”, “Connection Type”, “3GPP MMS Version”, “MM7 Schema” and “Remove White Space from MM7 XML” can all be left at blank or default values as they will be ignored for this type of connection.

“Max Connections” should be left blank or set to 1 if you are using our example script. (Our example script may become confused if it is being run multiple times simultaneously.)

“Non-Standard MM7 Variations” should be set to “PHP Multipart File Upload”.

The above definition only tells NowSMS how it can connect to your PHP script. However, it does not configure NowSMS to actually call the PHP script for the processing of received MMS messages. Completing this process will vary based upon how you receive MMS messages. There are three different configuration alternatives:

1.) MMS messages are received via a GSM/GPRS modem. In this configuration, there is an “MMS Settings” option under “Properties” for your GSM/GRPS modem definition in the “SMSC” list of NowSMS. The “MMS Settings” dialog includes an option titled “MMSC Routing for Received Messages” which should be set to “Route to MM7” with the name assigned to your PHP script selected (from the “MMSC Routing” definition that we completed in the previous step).

2.) MMS messages are received via a direct connection to an operator MMSC using one of the supported protocols, including MM7, MM4 or EAIF. When any of these protocols are used, the operator MMSC will automatically connect to your Now SMS/MMS Gateway to deliver messages. An “MMSC VASP” account is defined for the mobile operator connection on your NowSMS installation. The “MMSC VASP” definition includes an option titled “MMSC Routing for Received Messages” which should be set to “Route to MM7” with the name assigned to your PHP script selected (from the “MMSC Routing” definition that we completed in the previous step).

3.) NowSMS is the MMSC. In this case, you can use the “Route messages to this account for recipient phone number(s)” option in the “MMSC Routing” definition to route messages to your PHP script based upon the recipient phone number. (For example, define 1234 in this field, and if any of your MMSC users send an MMS message to 1234, it will be routed to your PHP script.)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved

For more information on operator MMSC connections and GSM/GPRS modems, please see the section entitled “Connecting to an Operator MMSC” in the NowSMS manual. (Also available on-line at http://www.nowsms.com/documentation/ProductDocumentation/mms_notifications_and_content/Connecting_to_operator_MMSC.htm.)

© Copyright 2002-2009 by NowMobile.com Limited, All Rights Reserved