



# **SMS Mobile Termination (MT) API Guide for HTTPS**

The content of this document are copyright and remain the property of National Informatics Centre. This document is not to be reproduced in any form whether electronic, mechanical or by other means without written permission of NIC.

© Copyright National Informatics Centre.

# Contents

---

1. Preface .....	4
1.1. Intended audience .....	4
1.2. Assumptions .....	4
1.3. Acronyms used in the guide.....	4
1.4. Convention used in the guide.....	5
1.5. Revision history .....	<b>Error! Bookmark not defined.</b>
2. Prerequisites.....	6
3. Introduction.....	6
4. API configurable parameters .....	7
5. Transmission of SMS.....	9
5.1. Normal text message .....	14
5.2. Text message of length more than 160 characters .....	15
5.3. Unicode characters.....	15
5.4. Binary characters.....	16
5.5. Error codes .....	18
6. Extended characters in GSM encoding .....	19
6.1. Double character space.....	19
6.2. Single character space.....	20
7. Delivery reports .....	24
7.1. Receiving delivery reports .....	25
7.2. Values passed to the URL.....	26
8. Sample scenario of events that occur when an SMS is sent.....	27
Glossary .....	28
Index .....	29
Additional Support .....	30

# List of tables

---

Table 1: Acronyms .....	4
Table 2: Conventions.....	5
Table 3: Configurable parameters.....	8
Table 4: PHP functions.....	12
Table 5: PHP function parameters.....	12
Table 6: Normal message – Input Parameters .....	14
Table 7: Message length more than 160 characters–Input Parameters .....	15
Table 8: Unicode messages– Input Parameters.....	16
Table 9: Binary messages - Input Parameters .....	17
Table 11: Transmission error codes .....	18
Table 12: Double character space in GSM encoding.....	20
Table 13: Single character space in GSM encoding .....	24
Table 14: Receiving delivery reports .....	25
Table 15: Values passed by NIC to the delivery URL.....	26

# 1. Preface

---

This API guide provides specifications of the HTTPS based APIs developed at NIC to send automated bulk SMS via the Internet. This messaging service uses the NIC gateway to send bulk SMSes. The NIC gateway not only sends bulk SMS but also sends delivery reports to its customers, based on the requirements.

## 1.1. Intended audience

---

This guide is intended for developers who plan to integrate their systems with NIC SMS service.

## 1.2. Assumptions

---

The guide assumes that you are familiar with and have sufficient experience in applying the following technologies and concepts:

- HTTPS communications using the GET and POST methods for parameter passing.
- Programming language such as Java, C/C++, Visual Basic or PHP.

## 1.3. Acronyms used in the guide

---

Acronyms	Description
ERR CD	Error Code
HLR	Home Location Register
MS	Mobile Station (Handset)
MSC	Mobile Switching Center
NIC	National Informatics Centre
PLMN	Public Land Mobile Network
SMSC	Short Message Service Center
VLR	Visitor Location Register

**Table 1: Acronyms**

## 1.4. Convention used in the guide

---

Convention	Description
<b>Bold font</b>	Used to provide emphasis for titles, captions and examples
Blue font	Used to represent cross references
Courier new font	Used to represent codes
Green font	Used to represent code comments
Red font	Used to represent notes
<i>Italics</i>	Used to represent important input data

**Table 2: Conventions**

## 2. Prerequisites

---

To use this HTTPS based API, you need to have a system with Internet connectivity on outbound port 443 (HTTPS). This connectivity may be direct to the Internet, or via a firewall or proxy server.

If you wish to receive delivery reports (DLRs) from the NIC system, you must have a web server that is able to receive incoming requests from the Internet on port 443 (HTTPS). This web server need not run on the same machine that sends messages to NIC.

## 3. Introduction

---

NIC introduces its new Enterprise Messaging Solution (EMS), the Bulk Push SMS. Companies can use this Bulk SMS service to send and receive messages across the world. The features of this service include:

- Sending bulk SMS by uploading a text message and an excel sheet of all destination mobile numbers.
- Masking sender ID - branding is one of the primary benefits of this feature.. Sender ID is masked so that users do not reply to the receiving number. The sender ID is masked to a ten digit number, an alphanumeric, or the company name.
- Provisioning for two-way messaging.
- Sending delivery reports that give the status of all the messages pushed.

The HTTPS APIs mentioned in the guide can be configured / integrated with any website/application to send bulk messages.

NIC Bulk Push SMS service is based on HTTPS. This SMS service can be used to send more than 100 SMS per second. This service can be integrated to perform two functions:

- Sending bulk SMS
- Checking the status of the SMS

## 4. API configurable parameters

The following table gives a list of API parameters that need to be set while creating API calls for distinct message services:

Name	Parameter	Description	Required Value	Default Value	Type
Username	<b>uname</b>	User name assigned to the account	Username		Required
Password	<b>pass</b>	Password assigned to the account	Password		Required
SenderID	<b>send</b>	Source Address of the message	Sender ID as required		Required
Destination Number	<b>dest</b>	Destination Address/Mobile Number of the message	Destination Mobile Number		Required
Message	<b>msg</b>	Text Content of the message (max length should not exceed 459 characters (if concatenation is on) or 160 characters for an 8 bit message). 280 characters for binary and 70 for Unicode characters.	Body of the message		Required
Priority	<b>prty</b>	The priority of the message being sent. The SMSC processes the message based on this priority value. If this value is not set in the URL, then the application takes the default value set at the account level.	1, 2, 3, 4, 5	Account Specific	Optional

Name	Parameter	Description	Required Value	Default Value	Type
Validity Period	<b>vp</b>	The time period after which the validity of the message expires. This can be between 1 to 180 minutes.	1 to 180 minutes	180	Optional
Schedule Time	<b>Schtm</b>	Set the value of scheduled time	22-08-2009 14:25 (dd-mm-YYYY H:m)	0	Optional
Concatenation	<b>concat</b>	Set this value to send multipart messages to the mobile.	0 = For limiting length to 160 char  1 = For concatenation	0	Optional
User Data Header	<b>udhi</b>	Set this value to indicate that the message payload is a binary message	0 = For normal text message  1 = For binary	0	Optional
Data Coding Scheme	<b>dcs</b>	Indicates the DCS value for the message. The 0 value indicates that the message payload is an 8 bit data.	0,4,8,240,245	0	Optional
Delivery URL	<b>dlr-url</b>	Set the value to 1, if you require delivery status report) call back.	0,1	0	Optional

**Table 3: Configurable parameters**

**Note:** To send a message to multiple destination numbers, enter multiple destination numbers separated by a comma under *dest* parameter. Here, maximum 100 mobile numbers can be put.

## 5. Transmission of SMS

NIC provides the HTTPS SMS Gateway that acts as the middleware component to transmit bulk SMS using the HTTPS URL that can be easily integrated with your application. For each successful submission, the HTTPS API returns a unique Message ID, which can be used to track the status of the message. Following is a sample API call that generates a Message ID:

<b>Input URL</b>	<code>https://smmsgw.nic.in/sendsms_nic/sendmsg.php?uname=xxx1&amp;pass=xxx2&amp;send=xxx3&amp;dest=xxx4&amp;msg=xxx5</code>
------------------	--

Here, in this URL, there are few important points you must keep in your mind without which you will not be able to send messages and instead you will get errors and you will not get any Message ID. Message ID acknowledges that SMS has been successfully sent from SMSGateway. If you do not get Message ID, it simply implies that you are doing something wrong. So, give proper attention to the following points to write HTTPS API correctly.

- 1) In the above URL, you see that `uname=xxx1`. Here, `xxx1` is nothing but the application ID which will be provided to you to use SMSGateway. This application ID is LDAP authenticated and is given permission to use SMSGateway by SMSGateway administrator. From any other IDs, you can not call the above URL and you will get errors.
- 2) In the above URL, you see that `pass=xxx2`. Here, `xxx2` is the password which you are given from our end along with application ID. You can change the password anytime in the same way you change the the password for your UID through which you log in to mail.nic.in.

**NOTE:** User is recommended to not change his password without intimating [smssupport@nic.in](mailto:smssupport@nic.in). And, any misuse of SMS service will result in inactivation of id.

a. To change the password given to you follow the following steps:-

- 1) Log in to mail.nic.in.
- 2) Go to Option tab, left click and then go to Global and left click.
- 3) Go to change password tab, left click it and then put the password provided by us in the Current Password and in the other two text boxes, write the password you want to have strictly following the password policy.

**Note:** Please use only "@" or "\*" or both while changing the password because if you use other special characters then you will have to convert that special character in the format acceptable by HTTPS URL. HTTPS URL does not take other special characters and white spaces directly. In the following pages(18-22), there is a detailed list of special characters and a decimal equivalent which need to be converted in Hexadecimal equivalent preceding with % before putting it into the HTTPS URL.

To see the list of characters and their decimal equivalent, please go to the link:-  
[Extended characters in GSM encoding](#)

For example: if my password is : abc#

Then, since # is a special character which can not be put directly in the HTTPS URL and if you see the list, you will see # is mapped to 35. So, first of all I will convert 35 in its hexadecimal equivalent which is 23. Then, I will have to precede it with %. So my password becomes abc%23, which I can put in the HTTPS URL.

- 3) In the above URL, next is send=xxx3, here, xxx3 is the sender ID which we will provide to you.
- 4) Then comes dest=xxx4, where you will have to put 10 digit mobile number preceding with 91. For ex: - If your mobile number is 9898123456 then you need to enter 919898123456 in dest parameter.
- 5) And, in msg=xxx5, xxx5 is the content of the message which you want to send.

**Note:** To send a message to multiple destination numbers, enter multiple destination numbers separated by a comma under *dest* parameter. Here, maximum 100 mobile numbers can be put.

When you call (HTTPS request) this API, a Message ID (MID) (for example, N7fab1c9b1a59e9f33ee1b7) is generated for each of the messages you submit. This MID is passed while forwarding the delivery report to your URL.

The HTTPS API provides the following messaging services:

- Normal text messages
- Text messages of length more than 160 characters
- Messages with Unicode characters
- Messages with binary characters

**Note:** To send a message scheduled for future, pass *sctm=22-08-2009 14:25* (dd-mm-YYYY H:m) along with the URL.

### PHP code

PHP is a scripting language that is widely used in designing dynamic web pages. PHP scripts are created using any text editor and embedded into HTML, which is then run on a web server. The PHP code is enclosed within the PHP tags. The PHP file is saved with a .php file extension.

Following is the **PHP** code snippet used to call this API:

```
// Initialize the text variable
$text=urlencode("xxx");

// Initialize the sender variable
$sender=urlencode("xxxxx");

// Initialize the URL variable
$url="https://smsgw.nic.in/sendsms_nic/sendmsg.php";

// Create and initialize a new cURL resource
$ch = curl_init();

// Set URL to URL variable
curl_setopt($ch, CURLOPT_URL,"https://$url");

// Set URL HTTPS post to 1
curl_setopt($ch, CURLOPT_POST, 1);

// Set URL HTTPS post field values
curl_setopt($ch, CURLOPT_POSTFIELDS,
"uname=xxx&pass=xx&send=xx&dest=xx&msg=xx");

// Set URL return value to True to return the transfer as a string
curl_setopt($ch, CURLOPT_RETURNTRANSFER, true);

// The URL session is executed and passed to the browser
$curl_output =curl_exec($ch);
```

The following table gives the PHP function details:

Function	
curl_init()	<b>Syntax:</b> resource curl_init ([ string \$url = NULL ] )
	<b>Description:</b> Initializes a new session and returns a cURL handle with the curl_setopt(), curl_exec(), and curl_close() functions.
	<b>Parameter:</b> URL
	<b>Return values:</b> Returns a cURL handle on success, or FALSE on errors.
curl_setopt()	<b>Syntax:</b> bool curl_setopt ( resource \$ch , int \$option , mixed \$value )

	<b>Description:</b> Sets options on the given cURL session handle.
	<b>Parameters:</b> ch (URL resource), option, value
	<b>Return values:</b> Returns TRUE on success or FALSE on failure.
curl_exec()	<b>Syntax:</b> mixed curl_exec ( resource \$ch )
	<b>Description:</b> Executes the given cURL session. This function should be called after initializing a cURL session and setting all the options for the session.
	<b>Parameters:</b> ch (URL resource)
	<b>Return values:</b> Returns TRUE on success or FALSE on failure. However, if the CURLOPT_RETURNTRANSFER option is set, it will return the result on success or FALSE on failure.

**Table 4: PHP functions**

The following table gives the curl\_setopt() function parameter option details:

Parameter	Description
CURLOPT_URL	To set the URL option.
CURLOPT_POST	To do a regular HTTPS Post.
CURLOPT_POSTFIELDS	To do a regular HTTPS field Post.
CURLOPT_RETURNTRANSFER	To return the transfer as a string for the return value of curl_exec().

**Table 5: PHP function parameters**

**.Net code**

The .Net code is written in the .Net Framework. The .Net Framework is an integral Windows component that supports development of applications and web services.

Following is the .Net code snippet used to call this API

```

// Create a new web request for the URL.
HttpWebRequest request = (HttpWebRequest)WebRequest.Create
(https://msgw.nic.in/sendsms_nic/sendmsg.php?..... );
/* Assign the response object of 'WebRequest' to the 'WebResponse'
variable, response.*/

```

```

HttpWebResponse response = (HttpWebResponse)request.GetResponse();
// Assign the response version to the String variable.
String ver = response.ProtocolVersion.ToString();
/* Create and assign the response stream to the StreamReader
variable.*/
StreamReader reader = new
StreamReader(response.GetResponseStream() );

```

Following is a detailed explanation of the classes used in the above API:

- **HttpRequest**

**Class description:** The `HttpRequest` class supports the properties and methods defined in `WebRequest` class and for additional properties and methods that enable the user to interact directly with servers using HTTPS.
  
- **WebRequest**

**Class description:** Makes a request to a Uniform Resource Identifier (URI). This is an abstract class for the .NET Framework's request/response model for accessing data from the Internet

**Example:** `HttpRequest req = (HttpRequest)WebRequest.Create(url);`

**Methods:**

  - `Create`: Initializes a new `WebRequest`.
  - `GetResponse`: When overridden in a descendant class, returns a response to an Internet request.
  
- **HttpWebResponse**

**Class description:** The `HttpWebResponse` class supports the HTTPS-specific uses of the properties and methods of the `WebResponse` class. The `HttpWebResponse` class is used to build HTTPS stand-alone client applications that send HTTPS requests and receive HTTPS responses.
  
- **WebResponse**

**Class description:** The `WebResponse` class is the abstract base class from which protocol-specific response classes are derived.

**Example:** `HttpWebResponse r =(HttpWebResponse)req.GetResponse();`

**Methods:**

  - `ToString`: Returns a `String` that represents the current `Object`.
  - `GetResponseStream`: When overridden in a descendant class, returns the data stream from the Internet resource.

## 5.1. Normal text message

---

While sending a normal 160 character message, do not use any optional parameter (except for dlr-url for receiving delivery acknowledgment on a predefined URL). Following is a sample API call that sets the message priority level to 3 and the validity period to 30 minutes.

<b>Input URL</b>	https://msgw.nic.in/sendsms_nic/sendmsg.php?uname=xxx1&pass=xxx2&send=xxx3&dest=xxx4&msg=xxx5&prty=3&vp=30
------------------	--

**Note:** In the above URL, replace xxx1, xxx2, xxx3, xxx4 and xxx5 by application ID, password of that ID (for LDAP authentication), sender ID, mobile number and the message (Actual message which you want to send). Please, look into page 9 and 10 for better explanation. These all parameters must be correct otherwise you will get errors.

For further explanation, go to, [Transmission of SMS](#)

**Here, message priority level is 3 and validity period of message is 30 minutes.** So in the URL, we have prty=3 and vp=30

Input Parameter	Example value
uname	XYZ
pass	ABC
send	NIC
dest	919945016016
msg	Testing
prty	3
vp	30

**Table 6: Normal message – Input Parameters**

For parameter description, see .

## 5.2. To send SMS in Bulk using HTTPS API

---

If user wants to send very large number of SMSs using HTTPS API then he can send SMS to maximum 100 numbers using the HTTPS URL discussed above. But what if the SMS needs to send to the lakhs of numbers, then he needs to call HTTPS URL from within a loop. The loop can be a for, while or do-while loop depending upon the requirement. He needs to fetch the number from a Database or from a flat file or whatever can be configured in the application and then the number must be put in the dest parameter in the HTTPS URL. He can use the php and .Net code of this document to call the HTTPS API.

Formatted: Bullets and Numbering

### 5.3. Text message of length more than 160 characters

The HTTPS API supports SAR (Segmentation and Reassembly – also called as concatenated SMS) of messages. The API supports a maximum of 3 segmentations with a maximum character limit of 459. To send concatenated SMS, set 'CONCAT' parameter value to 1. Following is a sample API call:

<b>Input URL</b>	https://msgw.nic.in/sendsms_nic/sendmsg.php?uname=xxx1&pass=xxx2&send=xxx3&dest=xxx4&msg=xxx5&concat=1
------------------	--

**Note:** In the above URL, replace xxx1, xxx2, xxx3, xxx4 and xxx5 by application ID, password of that ID (for LDAP authentication), sender ID, mobile number and the message (Actual message which you want to send). Please, look into page 9 and 10 for better explanation. These all parameters must be correct otherwise you will get errors.  
For further explanation, go to, [Transmission of SMS](#)

Input Parameter	Example value
Uname	XYZ
Pass	ABC
Send	NIC
Dest	919945016016
Msg	(Message more than 160 characters)
Concat	1

**Table 7: Message length more than 160 characters–Input Parameters**  
For parameter description, see .

Formatted: Bullets and Numbering

### 5.4. Unicode characters

NIC HTTPS API has an end-to-end support for Unicode characters. To send a Unicode message, set DCS value to 8 and the message text length to a maximum of 70 characters. Following is a sample API call:

<b>Input URL</b>	https://smmsgw.nic.in/sendsms_nic/sendmsg.php?uname=xxx1&pass=xxx2&send=xxx3&dest=xxx4&msg=xxx5&udhi=1&dc=8
------------------	---

**Note:** In the above URL, replace xxx1, xxx2, xxx3, xxx4 and xxx5 by application ID, password of that ID (for LDAP authentication), sender ID, mobile number and the message (Actual message which you want to send). Please, look into page 9 and 10 for better explanation. These all parameters must be correct otherwise you will get errors.  
 For further explanation, go to, [Transmission of SMS](#)

Input Parameter	Example value
Uname	XYZ
Pass	ABC
Send	NIC
Dest	919945016016
Msg	BinaryString
Udhi	1
Dcs	8

**Table 8: Unicode messages– Input Parameters**

For parameter description, see .

Formatted: Bullets and Numbering

## 5.5. Binary characters

HTTPS API supports 8-bit binary messages such as ring tones, logos, picture messages, and so on. To send a binary message the following parameters have to be set:

- UDH should be set to 1
- DCS value should be 4 or 240 or 245 (based on the spec)
- Message should not exceed 280 characters

Following is a sample API call:

<b>Input URL</b>	https://smmsgw.nic.in/sendsms_nic/sendmsg.php?uname=xxx1&pass=xxx2&send=xxx3&dest=xxx4&msg=xxx5&udhi=1&dc=4
------------------	---

**Note:** In the above URL, replace xxx1, xxx2, xxx3, xxx4 and xxx5 by application ID, password of that ID (for LDAP authentication), sender ID, mobile number and the message (Actual message which you want to send). Please, look into page 9 and 10 for better explanation. These all parameters must be correct otherwise you will get errors.

For further explanation, go to, [Transmission of SMS](#)

Input Parameter	Example value
Uname	XYZ
Pass	ABC
Send	NIC
Dest	919945016016
Msg	BinaryMessageString
Udhi	1
Dcs	4

**Table 9: Binary messages - Input Parameters**

For parameter description, see .

## 5.6. Specifications to receive (pull) messages:

---

The user will have to create a web page with the capabilities to capture parameters passed in its URL. We will forward the incoming message details as parameters to the URL. This can be achieved by capturing the query string.

Parameters:

Send = The mobile number from which the message is initiated.

Dest = The mobile number to which the message was sent.

Msg = The exact text of the message that was sent.

Stime = The time when the message was sent.

For Instance, the user creates a page xyz.aspx in their domain [www.abc.com](http://www.abc.com). We will forward a message by hitting the following URL:

<https://www.abc.com/xyx.aspx?send=919999999999&dest=919888888888&msg=Tesing&stime=yyyy-mm-dd 00:59:09>

**Note:** The user can create multiple keywords and configure URLs accordingly.

## 5.7. Error codes

The following table gives a list of error codes and their descriptions:

Error Codes	Error Description
0x200	Invalid Username or Password.
0x201	Account Suspended due to some reason.
0x202	Invalid Source Address/Sender Id. As per GSM standard the sender ID should be within 11 characters.
0x203	Message Length Exceeded(more than 160 chars) if concat is set to 0
0x204	Message Length Exceeded (more than 459 chars) if concat is set to 1.
0x205	DLR URL is not set.
0x206	Only the subscribed service type can be accessed, so ensure that the service type you are trying to connect with is a subscribed service type.
0x207	Invalid Source IP. Kindly check if the IP is responding.
0x208	Account Deactivated/Expired.
0x209	Invalid Message Length (less than 160 chars) if concat is set to 1.
0x210	Invalid Parameter values.
0x211	Invalid Message Length (more than 280 chars).
0x212	Invalid Message Length.
0x213	Invalid Destination number.

**Table 10: Transmission error codes**

## 6. Extended characters in GSM encoding

According to GSM specification, a standard SMS message can contain up to 160 characters (140 bytes). So while sending a text message, it is necessary to know the number of characters the message contains. It may be noted that some characters are counted as TWO characters in GSM Encoding.

NIC provides added support for such extended characters on its SMSC connectivity. The following section explains the double character spaces in SMS with an example.

### 6.1. Double character space

Table 11 lists a few characters which take double character spaces in SMS content. A double character space creates a dispute between the SMS length that you normally see while submitting and the actual length that appears on SMSC. So if you send a SMS containing 160 characters with any of the characters given in the following table, the actual length of the message adds up to 161 which is equivalent to two messages.

**Example:**

**A Sample SMS Body of 160 Characters:** "SMS is a new age medium which can be effectively used in today's business with a Mob handset in almost every employee / individual's hands. NIC."

**Actual SMS Length:** The above SMS actually consists of 161 characters as the Curly Bracket "}", occupies 2 character spaces. So the actual length becomes 161 and hence two SMS.

The following table gives a list of some characters that take a double character space in SMS content:

SI No.	Character name	Character	GSM Encoding (Hex Value)
1	Open Curly Bracket	{	1B28
2	Close Curly Bracket	}	1B29
3	Backslash	\	1B2F
4	Open Box Bracket	[	1B3C
5	Tilde	~	1B3D
6	Close Box Bracket	]	1B3E
7	Pipe		1B40
8	Caret	^	1B14

9	Form Feed	<FF>	1B0A
10	Euro Sign	€	1B 65

**Table 11: Double character space in GSM encoding**

## 6.2. Single character space

The following table gives a list of characters that take a single character space in SMS content:

SI No.	Character name	Character	ISO-8859-1 DEC
1	COMMERCIAL AT	@	64
2	POUND SIGN	£	163
3	DOLLAR SIGN	\$	36
4	YEN SIGN	¥	165
5	LATIN SMALL LETTER E WITH GRAVE	È	232
6	LATIN SMALL LETTER E WITH ACUTE	É	233
7	LATIN SMALL LETTER U WITH GRAVE	Ù	249
8	LATIN SMALL LETTER I WITH GRAVE	Ì	236
9	LATIN SMALL LETTER O WITH GRAVE	Ò	242
10	LATIN CAPITAL LETTER C WITH CEDILLA	Ç	199
11	LATIN FEED		10
12	LATIN CAPITAL LETTER O WITH STROKE	Ø	216
13	LATIN SMALL LETTER O WITH STROKE	ø	248
14	CARRIAGE RETURN		13
15	LATIN CAPITAL LETTER A WITH RING ABOVE	Å	197
16	LATIN SMALL LETTER A WITH RING ABOVE	å	229
17	GREEK CAPITAL LETTER DELTA	Δ	
18	LOW LINE	–	95
19	GREEK CAPITAL LETTER PHI	Φ	

SI No.	Character name	Character	ISO-8859-1 DEC
20	GREEK CAPITAL LETTER GAMMA	Γ	
21	GREEK CAPITAL LETTER LAMBDA	Λ	
22	GREEK CAPITAL LETTER OMEGA	Ω	
23	GREEK CAPITAL LETTER PI	Π	
24	GREEK CAPITAL LETTER PSI	Ψ	
25	GREEK CAPITAL LETTER SIGMA	Σ	
26	GREEK CAPITAL LETTER THETA	Θ	
27	GREEK CAPITAL LETTER XI	Ξ	
28	LATIN CAPITAL LETTER AE	Æ	198
29	LATIN SMALL LETTER AE	æ	230
30	LATIN SMALL LETTER SHARP S (German)	ß	223
31	LATIN CAPITAL LETTER E WITH ACUTE	É	201
32	SPACE		32
33	EXCLAMATION MARK	!	33
34	QUOTATION MARK	"	34
35	NUMBER SIGN	#	35
35	CURRENCY SIGN		164(ISO-8859-1)
36	PERCENT SIGN	%	37
37	AMPERSAND	&	38
38	APOSTROPHE	'	39
39	LEFT PARENTHESIS	(	40
40	RIGHT PARENTHESIS	)	41
41	ASTERISK	*	42
42	PLUS SIGN	+	43
43	COMMA	,	44
44	HYPHEN-MINUS	-	45
45	FULL STOP	.	46
46	SOLIDUS (SLASH)	/	47
47	DIGIT ZERO	0	48
48	DIGIT ONE	1	49

SI No.	Character name	Character	ISO-8859-1 DEC
49	DIGIT TWO	2	50
50	DIGIT THREE	3	51
51	DIGIT FOUR	4	52
52	DIGIT FIVE	5	53
53	DIGIT SIX	6	54
54	DIGIT SEVEN	7	55
55	DIGIT EIGHT	8	56
56	DIGIT NINE	9	57
57	COLON	:	58
58	SEMICOLON	;	59
59	LESS-THAN SIGN	<	60
60	EQUALS SIGN	=	61
61	GREATER-THAN SIGN	>	62
62	QUESTION MARK	?	63
63	INVERTED EXCLAMATION MARK	!	161
64	LATIN CAPITAL LETTER A	A	65
65	LATIN CAPITAL LETTER B	B	66
66	LATIN CAPITAL LETTER C	C	67
67	LATIN CAPITAL LETTER D	D	68
68	LATIN CAPITAL LETTER E	E	69
69	LATIN CAPITAL LETTER F	F	70
70	LATIN CAPITAL LETTER G	G	71
71	LATIN CAPITAL LETTER H	H	72
72	LATIN CAPITAL LETTER I	I	73
73	LATIN CAPITAL LETTER J	J	74
74	LATIN CAPITAL LETTER K	K	75
75	LATIN CAPITAL LETTER L	L	76
76	LATIN CAPITAL LETTER M	M	77
77	LATIN CAPITAL LETTER N	N	78
78	LATIN CAPITAL LETTER O	O	79
79	LATIN CAPITAL LETTER P	P	80
80	LATIN CAPITAL LETTER Q	Q	81
81	LATIN CAPITAL LETTER R	R	82
82	LATIN CAPITAL LETTER S	S	83
83	LATIN CAPITAL LETTER T	T	84

SI No.	Character name	Character	ISO-8859-1 DEC
84	LATIN CAPITAL LETTER U	U	85
85	LATIN CAPITAL LETTER V	V	86
86	LATIN CAPITAL LETTER W	W	87
87	LATIN CAPITAL LETTER X	X	88
88	LATIN CAPITAL LETTER Y	Y	89
89	LATIN CAPITAL LETTER Z	Z	90
90	LATIN CAPITAL LETTER A WITH DIAERESIS	Ä	196
91	LATIN CAPITAL LETTER O WITH DIAERESIS	Ö	214
92	LATIN CAPITAL LETTER N WITH TILDE	Ñ	209
93	LATIN CAPITAL LETTER U WITH DIAERESIS	Ü	220
94	SECTION SIGN	§	167
95	INVERTED QUESTION MARK	¿	191
96	LATIN SMALL LETTER A	A	97
97	LATIN SMALL LETTER B	B	98
98	LATIN SMALL LETTER C	C	99
99	LATIN SMALL LETTER D	D	100
100	LATIN SMALL LETTER E	E	101
101	LATIN SMALL LETTER F	F	102
102	LATIN SMALL LETTER G	G	103
103	LATIN SMALL LETTER H	H	104
104	LATIN SMALL LETTER I	I	105
105	LATIN SMALL LETTER J	J	106
106	LATIN SMALL LETTER K	K	107
107	LATIN SMALL LETTER L	L	108
108	LATIN SMALL LETTER M	M	109
109	LATIN SMALL LETTER N	N	110
110	LATIN SMALL LETTER O	O	111
111	LATIN SMALL LETTER P	P	112
112	LATIN SMALL LETTER Q	Q	113
113	LATIN SMALL LETTER R	R	114
114	LATIN SMALL LETTER S	S	115
115	LATIN SMALL LETTER T	T	116

SI No.	Character name	Character	ISO-8859-1 DEC
116	LATIN SMALL LETTER U	U	117
117	LATIN SMALL LETTER V	V	118
118	LATIN SMALL LETTER W	W	119
119	LATIN SMALL LETTER X	X	120
120	LATIN SMALL LETTER Y	Y	121
121	LATIN SMALL LETTER Z	Z	122
122	LATIN SMALL LETTER A WITH DIAERESIS	Ä	228
123	LATIN SMALL LETTER O WITH DIAERESIS	Ö	246
124	LATIN SMALL LETTER N WITH TILDE	Ñ	241
125	LATIN SMALL LETTER U WITH DIAERESIS	Ü	252
126	LATIN SMALL LETTER A WITH GRAVE	À	224

**Table 12: Single character space in GSM encoding**

## 7. Delivery reports

---

Each time the NIC SMS gateway attempts to deliver an SMS to a mobile device, a Delivery Report (DLR) is generated against every MID. The delivery report indicates if the message was successfully delivered and if not, the reason for failure.

The messages that are sent using the NIC SMS gateway can be in one of the following status at any given time:

- **Delivered**
- **Pending**
- **Failure**

**Delivered:** If a message is successfully delivered to the mobile device a Final DLR indicating a successful delivery is generated.

**Pending:** If a message is not delivered because of a temporary failure (such as the handset is unreachable or switched off, or its memory is full), the message goes to a pending status. NIC will continue to attempt to deliver the message for up to 3 hours (depending on the recipient’s mobile operator).

A pending status is always followed by a Final DLR at a later stage once the outcome of the message is known.

**Failure:** If a message fails because of a permanent failure (for example, if the message is sent to a telephone number that is not in service, or to a subscriber who is barred, or if the maximum re-tries for delivery have been exceeded), then a Final DLR is generated and no further re-tries will be attempted.

## 7.1. Receiving delivery reports

You can receive DLRs from NIC for messages sent via the NIC SMS gateway. These delivery reports may be useful in tracking message success rates, and in detecting invalid or out of service mobile numbers.

If you request for DLR delivery, you need to provide a HTTPS URL that can be called by the NIC system via Internet to deliver the DLR to your system. A sample parameter URL for NIC to forward the delivery report to your Delivery Interface is as follows:

<b>Input URL</b>	YourIPAddress/yourgateway/MessageStatus.aspx?sid=%s&stime=%s&dtime=%s&status=%s&reason=%s
------------------	---

Input Parameter	Description
Sid	Message ID printed on browser when a message is sent
Dest	Destination to which message is sent
Stime	Time when the message was submitted for delivery. Format - HH.MM.SS(Example, 17:44:59)
Dtime	Time when the message was delivered. Format - HH.MM.SS(Example, (17:45:07))
Status	Status of the message: 1 - Delivered 2 - Failed reason - On failure of the delivery of the message 3 - Expired, Pending status will not be given.
Reason	Reason behind the delivered/failed/expired sms like delv, abst subs, memory expired

**Table 13: Receiving delivery reports**

NIC will call this URL whenever a delivery report is available, and pass the parameters to the URL. The [Values passed to the URL](#) section gives a list of values that NIC will be passing to your URL.

Alternatively, NIC will provide you the front-end where you can view and download the status report.

Following is the NIC URL to download the status report. You can log in using your Username and Password to view and download the status report.

<b>Input URL</b>	https://msgw.nic.in/NIC/
------------------	--------------------------

## 7.2. Values passed to the URL

The following table lists the values passed to the URL and their description:

Values	Description
000	Message is submitted to SMSC (You can track this message from its Retrial Period (rp) and intermediate status if it is failed on its first attempt).
001	Message is successfully delivered to MS.
002	<b>Permanent Failure Reasons:</b> <ul style="list-style-type: none"> <li>• Unknown subscriber</li> <li>• Call Barred</li> <li>• Unidentified subscriber</li> <li>• Tele-service not provisioned</li> <li>• Facility not supported</li> <li>• Illegal equipment</li> <li>• Illegal subscribe</li> <li>• Unexpected data value</li> </ul>
003	<b>Temporary Failure Reasons</b>
032	SMS-Delivery Failure-MemCapExc: MS rejects the message because memory capacity exceeds i.e. no memory is available to store the message.
031	Absent SubscriberRR: MS is switched off / Not reachable. Absent Subscriber: MS is switched off / Not reachable.
035	Time Out: Network congestion (MSC not responding to SMS-GMSC with in the defined time period)
034	Time out-SC: Network congestion between SC (Service center) and SMS-GMSC.
033	System Failure: PLMN rejects due to NW/protocol failure.
036	Subscriber Busy For MT-SM: Current location of the MS is in another MSC area. VLR of visited MSC is unable / failed to update HLR (of corresponding MS) due to NW congestion.

**Table 14: Values passed by NIC to the delivery URL**

## 8. Sample scenario of events that occur when an SMS is sent

---

The following scenario reflects the events that occur when a Normal text message is sent. The URL should contain the following input values:

- Username
- Password
- Sender ID
- Destination ID/number
- Text message
- Priority value
- Validity period

Once this text message is transmitted, a MID is generated for the transaction. If the input URL contains erroneous values, the gateway generates the corresponding error code. For error code description, see [Error code](#) section.

DLRs are sent to the URL provided by the company. The URL provides details of the message status.

- If the message is successfully delivered, 001 is passed to the company URL.
- If the message is not delivered, a permanent or temporary failure message is passed to the company URL.
  - The value 002 is passed in case of a permanent failure.
  - The value 003 is passed in case of a temporary failure.

For more details on the values passed to the URL, see [Values passed to the URL](#) section.

**Note:** The flow of events to transmit text messages are similar if the messages have the following:

- More than 160 characters
- Unicode characters
- Binary characters and

They only vary in the input values passed to the URL.

## Glossary

---

Terms	Description
API	Application Programming Interface defines how an application program can request for services from an OS
Bulk SMS	To send large number of SMS to different mobile handsets via internet.
HTTPS	Hyper Text Transfer Protocol
PHP	Hypertext Preprocessor
Unicode	It is a 16 bit ISO 10646 character set. It is a standard that supports characters used by a large number of languages worldwide.

# Index

---

<b>B</b>		single character ..... 17
bulk SMS ..... 8	<b>M</b>	message ID ..... 8
<b>C</b>		
configurable parameters ..... 6	<b>P</b>	pending delivery reports ..... 21
<b>D</b>		
delivered delivery reports ..... 21	<b>R</b>	receive delivery reports ..... 22
delivery reports ..... 21	<b>S</b>	
delivered ..... 21	send	
failed ..... 22	binary character ..... 13	
pending ..... 21	bulk SMS ..... 8	
<b>E</b>	normal text ..... 12	
error codes ..... 15	text greater than 160 ..... 12	
<b>F</b>	Unicode character ..... 13	
failure delivery reports ..... 22	<b>T</b>	
<b>G</b>	transmission error codes ..... 15	
GSM encoding ..... 16	NIC API	
double character ..... 16	configurable parameters ..... 6	

## Additional Support

---

For any Technical support or queries, please contact us by dropping a mail on [smssupport@nic.in](mailto:smssupport@nic.in).