

**Free Russian Bride
For Every Reader**

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

Introduction.....	3
SMI Format.....	3
Sample File: T.smi	3
Sample File Annotated: T.smi	4
SMI Version Comparison.....	5
SMI File Breakdown	7
SMS Part (Mainly the PDU format)	9
Number Types:	11
TON (Type Of Number):	11
NPI (Numbering Plan Identifier):.....	12
Guide To 7 Bit Character Encoding	14
7 Bit Character Encoding Alphabet	15
References & Further Reading.....	19
Summaries	19
Alphabets & 7Bit Encoding.....	20
Other.....	20

No part of this publication may be reproduced, stored in a retrieval system, or transmitted in any form, or by any means, electronic, mechanical, recording, photocopying or otherwise without the expressed written permission of the Author.

Siemens is a registered trademark of Siemens AG. (Siemens®)

Introduction

Since the early Siemens mobile phones, users have been able to take advantage of the mobile's on-board allocated SMS memory to free up valuable SIM card storage space.

When a text message arrives the phone will try to store it in the phone's allocated SMS memory, this appears to be around 200 allocated spaces for SMS parts. Should the phone's memory fill up, the phone will continue to store messages on the SIM card. Once the SIM card's capacity has been reached, the phone reports that it is full ("SIM full. Please delete old messages") and hence no further messages can be delivered.

On Siemens mobile phones with a file system present you can move SMS messages to the phone's shared memory; this, on the Siemens A65, is around 1.5Mb and is shared with ringtones, pictures and other files.

Once messages are stored on the file system (usually in the "SMS Archive") they are encoded in either SMI (Siemens Message Inbox) or SMO (Siemens Message Outbox) files.

As a proprietary file format created by Siemens, you would have expected Siemens to have thoroughly documented SMI/SMO files. Unfortunately, I and many others have yet to find any documentation referring to them. Knowledge has been acquired about these relatively unknown formats via reverse engineering

SMI Format

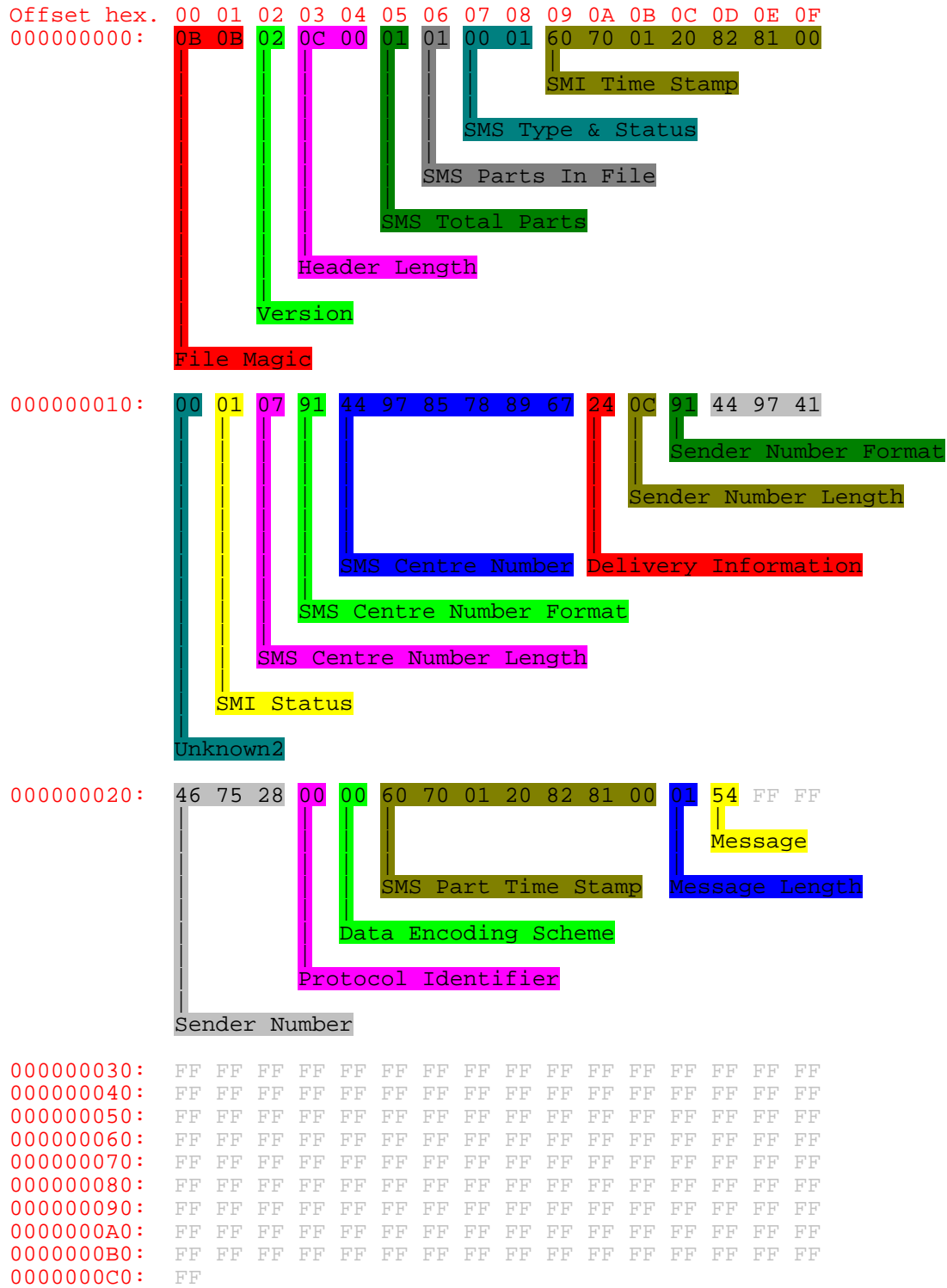
Sample File: T.smi

```
Offset hex. 00 01 02 03 04 05 06 07 08 09 0A 0B 0C 0D 0E 0F
00000000: 0B 0B 02 0C 00 01 01 00 01 60 70 01 20 82 81 00
00000010: 00 01 07 91 44 97 85 78 89 67 24 0C 91 44 97 41
00000020: 46 75 28 00 00 60 70 01 20 82 81 00 01 54 FF FF
00000030: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000040: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000050: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000060: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000070: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000080: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
00000090: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000A0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000B0: FF FF FF FF FF FF FF FF FF FF FF FF FF FF FF
000000C0: FF
```

FF = SMS Messages are padded so that the entire SMS is exactly 176 Bytes.

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

Sample File Annotated: T.smi



FF = SMS Messages are padded so that the entire SMS is exactly 176 Bytes.

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

SMI Version Comparison

(Adapted From <http://sisms.sourceforge.net/docs/SMISMOstruct.html>)

Field name & Length	Mobile phone model		
	Siemens SL4x (V0)	Siemens X45 (V1)	Siemens X55/X65/X75 (V2)
File Signature, Version & Header Length (5 bytes)	Yes	Yes	Yes
SMS Total & Included Parts (2 bytes)	No	Yes	Yes
SMS type (1 byte)	No	Yes	Yes
SMS status (1 byte)	No	Yes	Yes
Date/time stamp (7 bytes)	No	Yes	Yes
Unknown Byte (1 byte)	No	No	Yes
SMS Segment/part status/type (1 byte)	Yes	Yes	Yes
PDU structure (variable length)	Yes	Yes	Yes
Fill bytes (always 0xFF) (variable length)	Yes	Yes	Yes

An Introduction to SMI/SMO Files & The PDU Format V0.1 – NOT TO BE RE-DISTRIBUTED!

File features are as follows:

- **Version 0:** These files are the most simple in structure. They consist of the file signature, version and header length (0), SMS status/type and the PDU structure. They allow no SMS concatenation and have no date/time stamp except the date/time stamps of the corresponding file (creation, accessed and modified date).

Do not confuse the date/time stamp on the file with the message date/time stamp in later file versions. The file date/ time stamp is supported and operated by the file-system and is the date/time the SMI/SMO file was created (saved) to the archive. The message date/time stamp is recorded in the file itself and shows when the message was sent/received by the mobile phone.

Since file version 0 is supported by SL42/SL45, phones that do not support EMS, it is not expected that there will be EMS elements (font size, font type, pictures, etc.) present in these files.

Version 0 files are exactly 181 bytes long.

- **Version 1:** This version adds more information to the file.

It allows concatenated SMS messages. Additionally, there is support for EMS elements.

Version 1 adds PDU segments count, dedicated SMS type and SMS status bytes. It also features SMS message date/time stamp field (which is only set for SMI files; for SMO files it is 00s).

Version 1 file length varies according to the segments count in the message, but it is calculated with the formula $192 + 176 \times \text{Segment_count}$.

- **Version 2:** It is generally the same as version 1, but it has one more waste (unknown?) byte.

Message date/time stamp is present also for SMO files, but for mobiles later than Series 65.

Version 2 file length is calculated in the following way $193 + 176 \times \text{Segment_count}$.

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

SMI File Breakdown

Field	Description	Sample	Converted
- File Magic	Identifies that the file is an SMI/SMO file.	0B 0B	[NA]
- Version	Identifies what version of the SMI/SMO file format this is. 0 - Siemens SL4x compatible 1 - Siemens SL42 compatible 2 - Siemens X55 compatible	02	Version 2 (Siemens X55 Compatible)
- Header Length	Whether an SMI/SMO header exists (V1+) and the length of it in bytes.	0C 00	12 Bytes
- *SMS Total Parts	How many parts there are to this message.	01	01 Part(s)
- *SMS Included Parts	How many parts are included in this file.	01	01 Part(s)
- *SMS Type	The type of PDU enclosed. (Deliver/Submit) 00 – SMS-Deliver 03 – SMS-Submit	00	SMS-Deliver
- *SMS Status	The status of the entire SMS. <u>SMS-Deliver</u> 00 – Unread 01 – Read <u>SMS-Submit</u> 03 – Sent 04 - Unsent	01	Read
- *SMI Time Stamp	The file date and time stamp. (The numbers need to be reversed) The last integer specifies the number (in sign and magnitude notation) of 15 minutes to add to arrive at GMT. If this is negative then you subtract that number of 15 minutes. (E.g. -2 = -30 minutes)	60 70 01 20 82 81 00	06 07 10 02 28 18 00 YY-MM-DD HH:MM:SS +/- = 10 th July 2006, 02:28:18 GMT
- *Unknown	The meaning of this field is currently unknown. Note : Only In Version 2	00	[Unknown]
- SMS Part(s)	SMS Part Status & PDU (PDU Specified by ETSI) (Exactly 176 Bytes padded with FF's) Note : If the file version is Version 2 then the SMI can contain more than one SMS part	01 07 91 44 97 85 78 89 67 24 0C 91 44 97 41 46 75 28 00 00 60 70 01 20	[See Table Below]

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

		82 81 00 01 54 FF FF ..	
--	--	----------------------------------	--

SMS Part (Mainly the PDU format)

Field	Description	Sample	Converted
- SMS Status	Part Status SMI : 01 (Read), 03 (Unread) SMO : 05 (Sent), 07 (Unsent)	01	Read
- SMS PDU Part(s)	The next group of fields are copied from the SMS service centre exactly as they are received; this standard is the PDU standard and is documented by ESTI.		
- SMSC:Length	The length of the SMS Centre number. (Actual bytes if a number, number of characters if the type is alphanumeric) Important: Can Be 0!	07	07 Bytes
- *SMSC:Format	The type of the SMS Centre number. [See Table on Number Types]	91	International Number (+)
- *SMSC:Number	The actual SMS Centre number. (The numbers need to be reversed)	44 97 85 78 89 67	44 79 58 87 98 76 (Virgin Mobile)
- SMSC:Delivery Information	[See Table on Delivery Information]	24	[See Table]
- SMS Sender Length	The length of the SMS sender. (Actual bytes if a number, number of characters if the type is alphanumeric)	0C	12 Bytes
- SMS Sender Format	The type of the SMS Sender number. [See Table on Number Types]	91	International Number (+)
- SMS Sender Number	The number of the person who sent the SMS. (The numbers need to be reversed)	44 97 41 46 75 28	44 79 14 64 57 82
- Protocol ID	Which format the data is in. 00 - GSM 03,40	00	
- Data Coding Scheme	How the data is encoded. Bit 7 to 6 – Should the message delete itself? Bit 5 – Compression (1 True, 0 False) Bit 4 – Bit 1 to 0 have no meaning (1 True, 0 False) Bits 3 & 2 – Character Set	00	

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

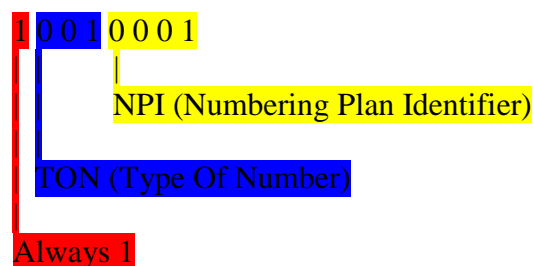
	Note : If all are 0000 0000 then the default GSM 7 bit alphabet with no message class is arisen.		
- SMS Centre Time Stamp	<p>The date/time the SMS part was received. (The numbers need to be reversed)</p> <p>The last integer specifies the number (in sign and magnitude notation) of 15 minutes to add to arrive at GMT. If this is negative then you subtract that number of 15 minutes.</p> <p>(E.g. -2 = -30 minutes)</p>	<p>60 70 01 20 82 81 00</p>	<p>06 07 10 02 28 18 00 YY-MM-DD HH:MM:SS +/-</p> <p>=</p> <p>10th July 2006, 02:28:18 GMT</p>
- Message Length	How many characters are in this part.	01	01 Character(s)
- Message	<p>The actual message encoded in 7 bits (unless otherwise specified) GSM encoding.</p> <p>(padded with 0xFF's to make up exactly 176 Bytes</p>	<p>54 FF FF ..</p>	T

* = Potentially optional depending on previous fields or SMI version number. E.g.

- SMI V0 doesn't have a header
- PDU's won't have a service centre if the service centre number length is 0

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

Number Types:



TON (Type Of Number):

Number	Description
000	Unknown (Everything else)
001	“International Number”, number begins with the national prefix (+)
010	“National Number”, number does not begin with a prefix.
011	“Network Specific Number”, e.g. Virgin Customer Services 789
100	Subscriber Number”, a short representation (possibly like short code numbers) should only be used in conjunction with the proper PID referring to this application
101	“Alphanumeric”, Coded in 7 bits.
110	“Abbreviated Number”
111	“Reserved”, For Future Use.

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

NPI (Numbering Plan Identifier):

Number	Description
0000	“Unknown” (Everything else)
0001	“ISDN/telephone numbering plan” (E.164/E.163)
0011	“Data numbering plan” (X.121).
0100	“Telex numbering plan” (often TTY machines for the deaf).
1000	“National numbering plan”
1001	“Private numbering plan”
1010	“ERMES numbering plan” (ETSI DE/PS 3 01-3)
1111	“Reserved”, For Future Use.

Note: The NPI is only for 000, 001 and 010 (Unknown, International and National) TON numbers, at all other times it will be 000

Note2: All other values are reserved.

Note3: For addressing the SC, MSC, SGSN or MS this will always be 0001, However to address the SME, anything can be used.

The most common value of this octet is 91 hex (10010001 bin), which indicates international format (with a +) a typical number would then be 46708251358 (where the country code is 46)

Another common value for this octet is 81 hex for numbers in the national format with a local prefix or numbers without prefix, here the same phone number would look like 0708251358.

The international format is the most generic, and it has to be accepted also when the message is destined to a recipient in the same country as the MSC or as the SGSN.

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

.. Left to complete :

- SMS Delivery Information
- Data Coding Scheme
- Protocol ID
- SMO
- SMS Header

An Introduction to SMI/SMO Files & The PDU Format
 V0.1 – NOT TO BE RE-DISTRIBUTED!

Guide To 7 Bit Character Encoding

(Adapted From <http://www.dreamfabric.com/sms/hello.html>)

The message "hellohello" consists of 10 characters, called septets when each character is represented by 7 bits each.

h	e	l	l	o	h	e	l	l	o
104	101	108	108	111	104	101	108	108	111
1101000	1100101	1101100	1101100	1101111	1101000	1100101	1101100	1101100	1101111
1101000	110010 1	11011 00	1101 100	110 1111	11 01000	1 100101	1101100 1	1101100	110111 1

- The first septet ('h') is turned into an octet by adding the rightmost bit of the second septet.
- This bit is inserted to the left which yields 1 + 1101000 = 11101000 ("E8").
- The rightmost bit of the second character is then consumed, so the second character (septet) needs two bits (yellow) of the third character to make an 8bit octet.
- This process goes on and on yielding the following octets:

1 1101000	00 110010	100 11011	1111 1101	01000 110	100101 11	1101100 1	1 1101100	110111
E8	32	9B	FD	46	97	D9	EC	37

The 9 octets from "hellohello" are "E8 32 9B FD 46 97 D9 EC 37"

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

7 Bit Character Encoding Alphabet

Hex	Character
0x00	@
0x01	£
0x02	\$
0x03	¥
0x04	è
0x05	é
0x06	ù
0x07	ì
0x08	ò
0x09	Ç
0x0A	
0x0B	Ø
0x0C	ø
0x0D	
0x0E	Å
0x0F	å
0x10	Δ
0x11	_
0x12	Φ
0x13	Γ
0x14	Λ
0x15	Ω
0x16	Π
0x17	Ψ
0x18	Σ
0x19	Θ
0x1A	Ξ
0x1B	
0x1B0A	
0x1B14	^
0x1B28	{
0x1B29	}
0x1B2F	\
0x1B3C	[
0x1B3D	~

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

0x1B3E]
0x1B40	
0x1B65	€
0x1C	Æ
0x1D	æ
0x1E	ß
0x1F	É
0x20	
0x21	!
0x22	"
0x23	#
0x24	¤
0x25	%
0x26	&
0x27	'
0x28	(
0x29)
0x2A	*
0x2B	+
0x2C	,
0x2D	-
0x2E	.
0x2F	/
0x30	0
0x31	1
0x32	2
0x33	3
0x34	4
0x35	5
0x36	6
0x37	7
0x38	8
0x39	9
0x3A	:
0x3B	;
0x3C	<
0x3D	=
0x3E	>

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

0x3F	?
0x40	i
0x41	A
0x42	B
0x43	C
0x44	D
0x45	E
0x46	F
0x47	G
0x48	H
0x49	I
0x4A	J
0x4B	K
0x4C	L
0x4D	M
0x4E	N
0x4F	O
0x50	P
0x51	Q
0x52	R
0x53	S
0x54	T
0x55	U
0x56	V
0x57	W
0x58	X
0x59	Y
0x5A	Z
0x5B	Ä
0x5C	Ö
0x5D	Ñ
0x5E	Ü
0x5F	§
0x60	ı
0x61	a
0x62	b
0x63	c
0x64	d

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

0x65	e
0x66	f
0x67	g
0x68	h
0x69	i
0x6A	j
0x6B	k
0x6C	l
0x6D	m
0x6E	n
0x6F	o
0x70	p
0x71	q
0x72	r
0x73	s
0x74	t
0x75	u
0x76	v
0x77	w
0x78	x
0x79	y
0x7A	z
0x7B	ä
0x7C	ö
0x7D	ñ
0x7E	ü
0x7F	à

References & Further Reading

Summaries

- “Kurznachrichten auf die perverse Art, http://www.nobbi.com/sms_pdu.htm
Nobbi (web.nobbi.01@nobbi.com)
Notes : General SMS PDU Summary
- “The GSM network offers a wireless infrastructure which extends your reach to anywhere in the world.”,
<http://www.usbdeveloper.com/GSMPage/gsmpage.htm>
Unknown (sale@usbdeveloper.com)
Notes : General SMS PDU Breakdown
- “SMS and the PDU format”, <http://www.dreamfabric.com/sms/>
Lars Pettersson (lars.pettersson@email.nu)
Notes : General SMS PDU Breakdown
- “Data Coding Scheme (TP-DCS)”, <http://www.dreamfabric.com/sms/dcs.html>
Lars Pettersson (lars.pettersson@email.nu)
Notes : TP-DCS Breakdown
- “Protocol Identifier (TP-PID)”, <http://www.dreamfabric.com/sms/pid.html>
Lars Pettersson (lars.pettersson@email.nu)
Notes : TP-PID Breakdown
- “Service Centre Time Stamp (TP-SCTS)”,
<http://www.dreamfabric.com/sms/scts.html>
Lars Pettersson (lars.pettersson@email.nu)
Notes : TP-SCTS Breakdown
- “First octet of the SMS-DELIVER PDU”,
http://www.dreamfabric.com/sms/deliver_fo.html
Lars Pettersson (lars.pettersson@email.nu)
Notes : SMS-Deliver
- “The Type-of-Address octet”,
http://www.dreamfabric.com/sms/type_of_address.html
Lars Pettersson (lars.pettersson@email.nu)
Notes : Type of Address Breakdown
- “Validity Period (TP-VP)”, <http://www.dreamfabric.com/sms/vp.html>
Lars Pettersson (lars.pettersson@email.nu)
Notes : Validity Period Breakdown
- “Siemens SMI/SMO File Structure”,
<http://sisms.sourceforge.net/docs/SMISMOStruct.html>
SiSMS Team (Unknown)
Notes : SMI Breakdown

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

- “Siemens .smi/.smi/SMS.dat file format”,
<http://www.hendrik-sattler.de/> (.txt file inside the scmxx package)
Hendrik Sattler (post@hendrik-sattler.de)
Notes : Good SMI Breakdown
- “SMS and PDU format”,
http://home.student.utwente.nl/s.p.ekkebus/portfolio/resource/sms_pdu.html
Swen-Peter Ekkebus (ekkebus@cs.utwente.nl)
Notes : Poor PDU Breakdown (But has a good JavaScript example)

Alphabets & 7Bit Encoding

- “The 7 bit representation of SMS characters”,
http://www.usbdeveloper.com/GSMPage/gsmpage_7bittable.htm,
Unknown (sale@usbdeveloper.com)
Notes : 7bit Character Alphabet
- “The 7 bit default alphabet”,
http://www.dreamfabric.com/sms/default_alphabet.html
Lars Pettersson (lars.pettersson@email.nu)
Notes : 7bit Character Alphabet (With ISO Decimal Equivalents)
- “Coding 7-bit data (septets) into octets”,
<http://www.dreamfabric.com/sms/hello.html>
Lars Pettersson (lars.pettersson@email.nu)
Notes : 7Bit Encoding Example

Other

- “+CMS ERROR”, http://www.dreamfabric.com/sms/cms_error.html,
Lars Pettersson (lars.pettersson@email.nu)
Notes : Message Status Error Codes
- “First octet of the SMS-SUBMIT PDU”.
http://www.dreamfabric.com/sms/submit_fo.html
Lars Pettersson (lars.pettersson@email.nu)
Notes : SMS-Submit
- “Flash SMS”, <http://www.dreamfabric.com/sms/alert.html>
Lars Pettersson (lars.pettersson@email.nu)
Notes : SMS Flash Message Class
- “I want to send messages to mobile phones.. what do I need?”,
<http://smsforum.net/smf/index.php?action=printpage;topic=2117.0>
SMS Forum Support (Unknown)
Notes : Details on the 3 ways to get your message out there.
- “SMS Messaging”, <http://www.activexperts.com/activsms/sms/technical/>
ActiveXperts Software (Unknown)
Notes : Good Acronym List

An Introduction to SMI/SMO Files & The PDU Format
V0.1 – NOT TO BE RE-DISTRIBUTED!

- “SMS Tutorial”, http://www.funsms.net/sms_tutorial.htm
funsms.net (Unknown)
Notes : Good History Information

Note: The Siemens AT Command reference is also a useful resource when programmatically accessing a mobile phone.

Note2: There are a LOT of mistakes on some of these pages (e.g. Stating there are only 2 types of number types, “SMS Type” field marked as “Unknown”, Ignoring the GMT field on the timestamp.. etc etc), this document has been devised to correct half of the inaccuracies expressed on websites, however check and test thoroughly.