3GPP TSG-T (Terminals) Meeting #15
Jeju Island, Korea, 6 - 8 March, 2002

Tdoc TP-020015

Agenda Item: 5.2.3

Source: T2

Title: "SMS/EMS/CBS" Change Requests

Document for: Approval

Spec	CR	Rev	Rel	Subject	Cat	Vers-Curr	Vers-New	T2 Tdoc	Workitem
23.038	009		Rel-5	User Data Header support over CBS	F	4.4.0	5.0.0	T2-020231	TEI5
23.040	041		Rel-5	Wireless Vector Graphics in EMS	B	5.2.0	5.3.0	T2-020301	MESS5-EMS
23.040	042		Rel-5	Polyphonic Extended Object	B	5.2.0	5.3.0	T2-020302	MESS5-EMS
23.040	043		R99	MO-SMS duplicate message response	F	3.7.0	3.8.0	T2-020225	TEI
23.040	044		Rel-4	MO-SMS duplicate message response	A	4.5.0	4.6.0	T2-020226	TEI4
23.040	045		Rel-5	MO-SMS duplicate message response	A	5.2.0	5.3.0	T2-020227	TEI5
23.040	046	1	Rel-5	Subaddressing scheme for SMS	B	5.2.0	5.3.0	T2-020304	TEI5
23.040	047		Rel-5	Alternate Reply Address Element	B	5.2.0	5.3.0	T2-020230	TEI5
23.040	048		Rel-5	Extended Object Data Request Command	C	5.2.0	5.3.0	T2-020237	MESS5-EMS

3GPP TSG-T2#16
Sophia-Antipolis, France
11-15 February 2002

T2-020231

CR-Form-v3

CHANGE REQUEST

⌘ **23.038 CR 009** ⌘ rev **-** ⌘ Current version: **4.4.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network □ Core Network □

Title:	⌘	User Data Header support over CBS			
Source:	⌘	T2			
Work item code:	⌘	TEI5	***Date:***	⌘	14-February 2002
Category:	⌘	**F**	***Release:***	⌘	Rel-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	There is no clear definition for sending messages over CB with a User Data Header Structure
Summary of change:	⌘	Define the specific DCS values that mention that CB message payload is encoded with a User Data Header structure.
Consequences if not approved:	⌘	Ambiguities in specifications may lead to interoperability issues.

Clauses affected:	⌘			
Other specs Affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

5 CBS Data Coding Scheme

The CBS Data Coding Scheme indicates the intended handling of the message at the MS, the alphabet/coding, and the language (when applicable). Any reserved codings shall be assumed to be the GSM 7 bit default alphabet (the same as codepoint 00001111) by a receiving entity. The octet is used according to a coding group which is indicated in bits 7..4. The octet is then coded as follows:

Coding Group Bits 7..4	Use of bits 3..0
0000	Language using the GSM 7 bit default alphabet Bits 3..0 indicate the language: 0000 German 0001 English 0010 Italian 0011 French 0100 Spanish 0101 Dutch 0110 Swedish 0111 Danish 1000 Portuguese 1001 Finnish 1010 Norwegian 1011 Greek 1100 Turkish 1101 Hungarian 1110 Polish 1111 Language unspecified
0001	0000 GSM 7 bit default alphabet; message preceded by language indication. The first 3 characters of the message are a two-character representation of the language encoded according to ISO 639 [12], followed by a CR character. The CR character is then followed by 90 characters of text. 0001 UCS2; message preceded by language indication The message starts with a two 7-bit default alphabet character representation of the language encoded according to ISO 639 [12]. This is padded to the octet boundary with two bits set to 0 and then followed by 40 characters of UCS2-encoded message. An MS not supporting UCS2 coding will present the two character language identifier followed by improperly interpreted user data. 0010..1111 Reserved
0010..	0000 Czech 0001 Hebrew 0010 Arabic 0011 Russian 0100 Icelandic 0101..1111 Reserved for other languages using the GSM 7 bit default alphabet, with unspecified handling at the MS
0011	0000..1111 Reserved for other languages using the GSM 7 bit default alphabet, with unspecified handling at the MS
01xx	General Data Coding indication Bits 5..0 indicate the following: Bit 5, if set to 0, indicates the text is uncompressed Bit 5, if set to 1, indicates the text is compressed using the compression algorithm defined in 3GPP TS 23.042 [13]

Coding Group Bits 7..4	Use of bits 3..0
	Bit 4, if set to 0, indicates that bits 1 to 0 are reserved and have no message class meaning Bit 4, if set to 1, indicates that bits 1 to 0 have a message class meaning: Bit 1 Bit 0 Message Class: 0 0 Class 0 0 1 Class 1 Default meaning: ME-specific. 1 0 Class 2 (U)SIM specific message. 1 1 Class 3 Default meaning: TE-specific (see 3GPP TS 27.005 [8]) Bits 3 and 2 indicate the alphabet being used, as follows: Bit 3 Bit 2 Alphabet: 0 0 GSM 7 bit default alphabet 0 1 8 bit data 1 0 USC2 (16 bit) [10] 1 1 Reserved
1000	Reserved coding groups
1001	Message with User Data Header (UDH) structure: Bit 1 Bit 0 Message Class: 0 0 Class 0 0 1 Class 1 Default meaning: ME-specific. 1 0 Class 2 (U)SIM specific message. 1 1 Class 3 Default meaning: TE-specific (see 3GPP TS 27.005 [8]) Bits 3 and 2 indicate the alphabet being used, as follows: Bit 3 Bit 2 Alphabet: 0 0 GSM 7 bit default alphabet 0 1 8 bit data 1 0 USC2 (16 bit) [10] 1 1 Reserved
1010~~00~~..1101	Reserved coding groups
1110	Defined by the WAP Forum [15]
1111	Data coding / message handling Bit 3 is reserved, set to 0. Bit 2 Message coding: 0 GSM 7 bit default alphabet 1 8 bit data Bit 1 Bit 0 Message Class: 0 0 No message class. 0 1 Class 1 user defined. 1 0 Class 2 user defined. 1 1 Class 3 default meaning: TE specific (see 3GPP TS 27.005 [8])

These codings may also be used for USSD and MMI/display purposes.

See 3GPP TS 24.090 [11] for specific coding values applicable to USSD for MS originated USSD messages and MS terminated USSD messages. USSD messages using the default alphabet are coded with the GSM 7-bit default alphabet given in subclause 6.2.1. The message can then consist of up to 182 user characters.

Cell Broadcast messages using the default alphabet are coded with the GSM 7-bit default alphabet given in subclause 6.2.1. The message then consists of 93 user characters.

If the GSM 7 bit default alphabet extension mechanism is used then the number of displayable characters will reduce by one for every instance where the GSM 7 bit default alphabet extension table is usedCell Broadcast messages using 8-bit data have user-defined coding, and will be 82 octets in length.

UCS2 alphabet indicates that the message is coded in UCS2 [10]. The General notes specified in subclause 6.1.1 override any contrary specification in UCS2, so for example even in UCS2 a <CR> character will cause the MS to return to the beginning of the current line and overwrite any existing text with the characters which follow the <CR>. Messages encoded in UCS2 consist of 41 characters.

Class 1 and Class 2 messages may be routed by the ME to user-defined destinations, but the user may override any default meaning and select their own routing.

Class 3 messages will normally be selected for transfer to a TE, in cases where a ME supports an SMS/CBS interface to a TE, and the TE requests "TE-specific" cell broadcast messages (see 3GPP TS 27.005 [8]). The user may be able to override the default meaning and select their own routing.

Messages with a User Data Header Structure are encoded as described in 3GPP TS 23.040 [4] for SMS, in subclauses 3.10 and 9.2.3.24.

The use of Cell Broadcast DCS values for messages with a User Data Header structure implies that the 82-bytes CB payload has a User Data Header structure.

The CBS message information field will contain the IEs as described in 3GPP TS 23.040. The concatenation IEs will not be used, as CB concatenation will rely in that case on the existing CB mechanism. Note that IEs that cannot be split and that IEs that are too large to fit in one CB segment cannot be transmitted using this mechanism. Also, some IEs as defined for SMS are not applicable for CB:

VALUE (hex)	MEANING
00	Concatenated short messages, 8-bit reference number
01	Special SMS Message Indication
06	SMSC Control Parameters
08	Concatenated short message, 16-bit reference number
20	RFC 822 E-Mail Header
70-7F	(U)SIM Toolkit Security Headers
80-89	SME to SME specific use

3GPP TSG-T2 #16
Sophia Antipolis, France
11-15 February 2002

T2-020237

CR-Form-v5

CHANGE REQUEST

⌘ **23.040 CR 048** ⌘**rev** **-** ⌘ Current version: **5.2.0** ⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network □ Core Network □

Title:	⌘	Extended Object Data Request Command
Source:	⌘	T2
Work item code:	⌘	MESS5-EMS
Date:	⌘	February 11, 2002
Category:	⌘	**C**
Release:	⌘	REL-5

Use one of the following categories:
F *(correction)*
A *(corresponds to a correction in an earlier release)*
B *(addition of feature),*
C *(functional modification of feature)*
D *(editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	The Data Format Delivery Request mechanism in the Extended Object Information Element communicates the data formats a terminal requests. While such a mechanism is positive to improve the lack of interoperability within EMS, it’s terminal based and not a full solution to the problem.
Summary of change:	⌘	The purpose of this CR is to include a new command sourced from an SMSC requesting the Data Format Delivery Request. Such a command will enable a network-based solution for interoperability of EMS.
Consequences if not approved:	⌘	Network Operators will not fully realize the potential revenue from services based on SMS, due to the lack of interoperability within EMS.

Clauses affected:	⌘	New Section 6.3, Section 9.2.3.24, New Section 9.2.3.24.10.1.15
Other specs affected:	⌘	□ Other core specifications ⌘ □ Test specifications □ O&M Specifications
Other comments:	⌘	

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6 Service Centre functionality

In the present document, only the SC functionality related to the short message service between the SC and the MS is specified.

6.1 Service Centre capabilities

The SC should be capable of:

- submitting a short message to an MS, retaining the responsibility of the message until

    1) the report has been received; or

    2) the Validity-Period expires.

- receiving a report from the PLMN;
- receiving a short message from an MS;
- returning a report to the PLMN for a previously received short message.

6.2 SC functional requirements

The detailed functionality of the SC is outside the scope of the present document, and is for the SC operator to define. However, the following functional requirements are mandatory for all SCs in order to support the SM-TP (see clause 9) towards the PLMN:

1) To identify each SMS-DELIVER sent to an MS in a unique way, a time stamp value is included in the field TP-Service-Centre-Time-Stamp, TP-SCTS, of the SMS-DELIVER. The time stamp gives the time when the message arrived at the SC with the accuracy of a second. If two or more messages to the same MS arrive at the SC within one second, the SC shall modify the time stamp of those messages in such a way that:

    a) all messages to the MS contain different time stamps;

    b) the modification of the time stamps is kept to a minimum.

2) The SC is only allowed to have one outstanding SMS-DELIVER (i.e. a message for which a report has not been received) to a specific MS at a given time.

3) The SC shall be able to initiate overwriting of short messages previously received by the SC if requested by the same originating address (MS or any other source) by use of the same message type.

6.3 SC EMS Extended Object Data Request Command Feature

An SME has the ability of determining which data formats within the Extended Object IE are supported by a specific terminal. The SC has the option of supporting this feature using an SMS-DELIVER PDU. This SMS-DELIVER PDU shall contain the EMS Data Format Delivery Request IE, and be marked for automatic deletion by the mobile station.

……

…….

……

…….

9.2.3.23 TP-User-Data-Header-Indicator (TP-UDHI)

The TP-User-Data-Header-Indicator is a 1 bit field within bit 6 of the first octet of the following six PDUs:

- SMS-SUBMIT,
- SMS-SUBMIT-REPORT,
- SMS-DELIVER,
- SMS-DELIVER-REPORT,
- SMS-STATUS-REPORT,
- SMS-COMMAND.

TP-UDHI has the following values.

Bit no. 6 0 The TP-UD field contains only the short message

1 The beginning of the TP-UD field contains a Header in addition to the short message.

9.2.3.24 TP-User Data (TP-UD)

The length of the TP-User-Data field is defined in the PDU’s of the SM-TL (see clause 9.2.2).

The TP-User-Data field may comprise just the short message itself or a Header in addition to the short message depending upon the setting of TP-UDHI.

Where the TP-UDHI value is set to 0 the TP-User-Data field comprises the short message only, where the user data can be 7 bit (default alphabet) data, 8 bit data, or 16 bit (UCS2 [24]) data.

Where the TP-UDHI value is set to 1 the first octets of the TP-User-Data field contains a Header in the following order starting at the first octet of the TP-User-Data field.

Irrespective of whether any part of the User Data Header is ignored or discarded, the MS shall always store the entire TPDU exactly as received.

FIELD	LENGTH
Length of User Data Header	1 octet
Information-Element-Identifier "A"	1 octet
Length of Information-Element "A"	1 octet
Information-Element "A" Data	0 to "n" octets
Information-Element-Identifier "B"	1 octet
Length of Information-Element "B"	1 octet
Information-Element "B" Data	0 to "n" octets
Information-Element-Identifier "X"	1 octet
Length of Information-Element "X"	1 octet
Information-Element "X" Data	0 to "n" octets

The diagram below shows the layout of the TP-User-Data-Length and the TP-User-Data for uncompressed GSM 7 bit default alphabet data. The UDHL field is the first octet of the TP-User-Data content of the Short Message.

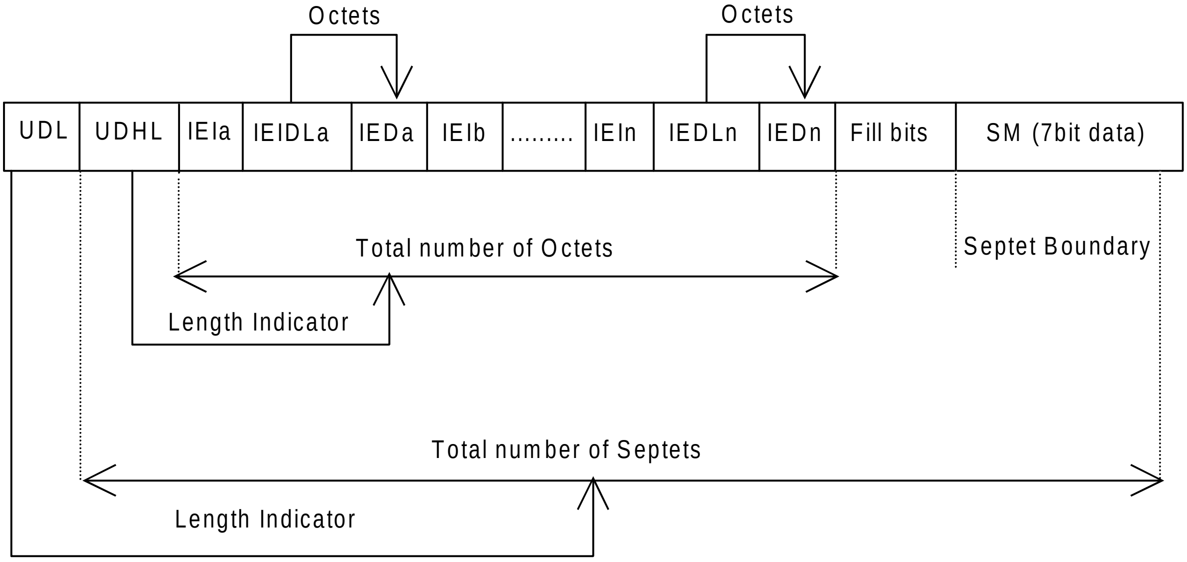


Figure 9.2.3.24 (a)

The diagram below shows the layout of the TP-User-Data-Length and the TP-User-Data for uncompressed 8 bit data or uncompressed UCS2 data. The UDHL field is the first octet of the TP-User-Data content of the Short Message.

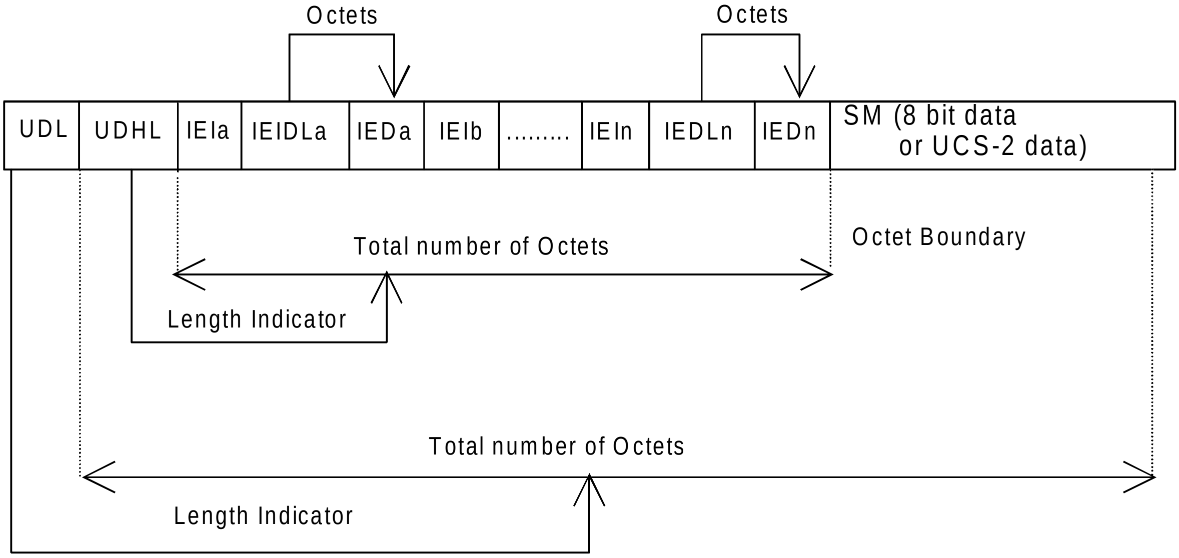


Figure 9.2.3.24 (b)

The diagram below shows the layout of the TP-User-Data-Length and the TP-User-Data for compressed GSM 7 bit default alphabet data, compressed 8 bit data or compressed UCS2 data. The UDHL field is the first octet of the TP-User-Data content of the Short Message.

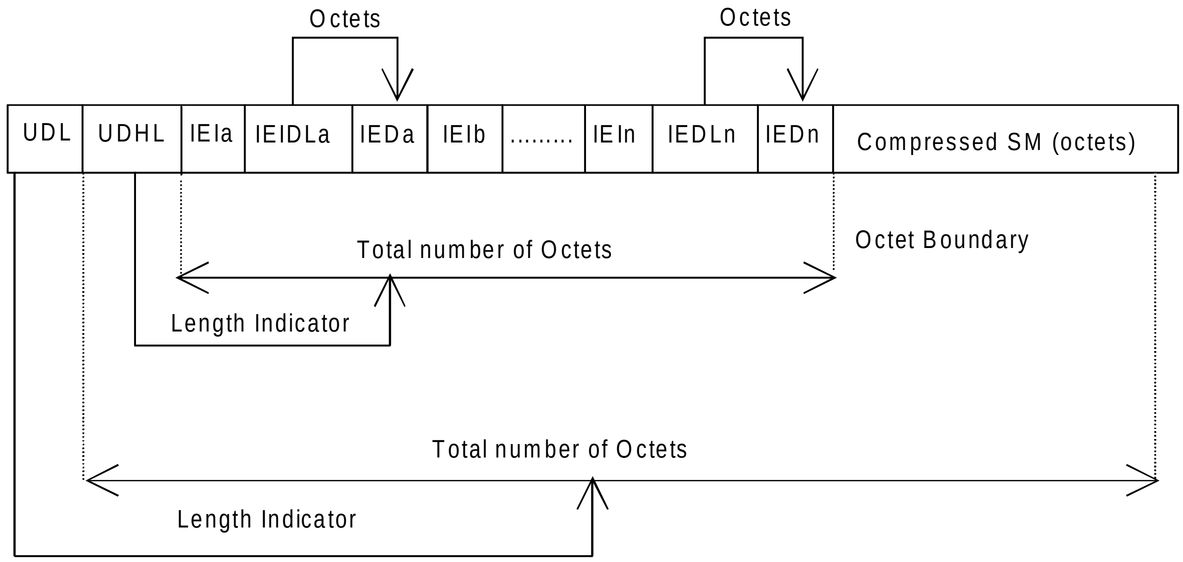


Figure 9.2.3.24 (c)

The definition of the TP-User-Data-Length field which immediately precedes the "Length of User Data Header" is unchanged and shall therefore be the total length of the TP-User-Data field including the Header, if present. (see 9.2.3.16).

The "Length-of-Information-Element" fields shall be the integer representation of the number of octets within its associated "Information-Element-Data" field which follows and shall not include itself in its count value.

The "Length-of-User-Data-Header" field shall be the integer representation of the number of octets within the "User-Data-Header" information fields which follow and shall not include itself in its count or any fill bits which may be present (see text below).

Information Elements may appear in any order and need not follow the order used in the present document. Information Elements are classified into 3 categories as described below.

- SMS Control – identifies those IEIs which have the capability of dictating SMS functionality.
- EMS Control – identifies those IEIs which manage EMS Content IEIs.
- EMS Content – identifies those IEIs containing data of a unique media format.

It is permissible for certain IEs to be repeated within a short message, or within a concatenated message. There is no restriction on the repeatability of IEs in the EMS Content classification. The repeatability of SMS Control and EMS Control IEs is determined on an individual basis. See the IE table below for the repeatability of each IE.

In the event that IEs determined as not repeatable are duplicated, the last occurrence of the IE shall be used. In the event that two or more IEs occur which have mutually exclusive meanings (e.g. an 8bit port address and a 16bit port address), then the last occurring IE shall be used.

If the length of the User Data Header is such that there are too few or too many octets in the final Information Element then the whole User Data Header shall be ignored.

If any reserved values are received within the content of any Information Element then that part of the Information Element shall be ignored.

The Information Element Identifier octet shall be coded as follows:

VALUE (hex)	MEANING	Classification	Repeatability
00	Concatenated short messages, 8-bit reference number	SMS Control	No
01	Special SMS Message Indication	SMS Control	Yes
02	Reserved	N/A	N/A
03	Value not used to avoid misinterpretation as <LF> character	N/A	N/A
04	Application port addressing scheme, 8 bit address	SMS Control	No
05	Application port addressing scheme, 16 bit address	SMS Control	No
06	SMSC Control Parameters	SMS Control	No
07	UDH Source Indicator	SMS Control	Yes
08	Concatenated short message, 16-bit reference number	SMS Control	No
09	Wireless Control Message Protocol	SMS Control	Note 3
0A	Text Formatting	EMS Control	Yes
0B	Predefined Sound	EMS Content	Yes
0C	User Defined Sound (iMelody max 128 bytes)	EMS Content	Yes
0D	Predefined Animation	EMS Content	Yes
0E	Large Animation (16*16 times 4 = 32*4 =128 bytes)	EMS Content	Yes
0F	Small Animation (8*8 times 4 = 8*4 =32 bytes)	EMS Content	Yes
10	Large Picture (32*32 = 128 bytes)	EMS Content	Yes
11	Small Picture (16*16 = 32 bytes)	EMS Content	Yes
12	Variable Picture	EMS Content	Yes
13	User prompt indicator	EMS Control	Yes
14	Extended Object	EMS Content	Yes
15	Reused Extended Object	EMS Control	Yes
16	Compression Control	EMS Control	No
17	Object Distribution Indicator	EMS Control	Yes
18	Extended Object Data Request Command	EMS Control	No
19~~8~~-1F	Reserved for future EMS features (see subclause 3.10)	N/A	N/A
20	RFC 822 E-Mail Header	SMS Control	No
21	Hyperlink format element	SMS Control	Yes
22 – 6F	Reserved for future use	N/A	N/A
70 – 7F	(U)SIM Toolkit Security Headers	SMS Control	Note 1
80 – 9F	SME to SME specific use	SMS Control	Note 2
A0 – BF	Reserved for future use	N/A	N/A
C0 – DF	SC specific use	SMS Control	Note 2
E0 – FF	Reserved for future use	N/A	N/A
Note 1: The functionality of these IEIs is defined in 3GPP TSG 23.048 [28], and therefore, the repeatability is not within the scope of this document and will not be determined here. Note 2: The functionality of these IEIs is used in a proprietary fashion by different SMSC vendors, and therefore, are not within the scope of this technical specification. Note 3: The functionality of these IEIs is defined by the WAP Forum and therefore the repeatability is not within the scope of this document and will not be determined here.			

A receiving entity shall ignore (i.e. skip over and commence processing at the next information element) any information element where the IEI is Reserved or not supported. The receiving entity calculates the start of the next information element by looking at the length of the current information element and skipping that number of octets.

The SM itself may be coded as 7, 8 or 16 bit data.

If 7 bit data is used and the TP-UD-Header does not finish on a septet boundary then fill bits are inserted after the last Information Element Data octet up to the next septet boundary so that there is an integral number of septets for the entire TP-UD header. This is to ensure that the SM itself starts on an septet boundary so that an earlier Phase mobile shall be capable of displaying the SM itself although the TP-UD Header in the TP-UD field may not be understood.

It is optional to make the first character of the SM itself a Carriage Return character encoded according to the default 7 bit alphabet so that earlier Phase mobiles, which do not understand the TP-UD-Header, shall over-write the displayed TP-UD-Header with the SM itself.

If 16 bit (USC2) data is used then padding octets are not necessary. The SM itself shall start on an octet boundary.

If 8 bit data is used then padding is not necessary. An earlier Phase mobile shall be able to display the SM itself although the TP-UD header may not be understood.

It is also possible for mobiles not wishing to support the TP-UD header to check the value of the TP-UDHI bit in the SMS-Deliver PDU and the first octet of the TP-UD field and skip to the start of the SM and ignore the TP-UD header.

9.2.3.24.1 Concatenated Short Messages

This facility allows short messages to be concatenated to form a longer message.

In the case of uncompressed 8-bit data, the maximum length of the short message within the TP-UD field is 134 (140-6) octets.

In the case of uncompressed GSM 7 bit default alphabet data, the maximum length of the short message within the TP-UD field is 153 (160-7) characters.

In the case of 16 bit uncompressed USC2 data, the maximum length of the short message within the TP-UD field is 67 ((140-6)/2) characters. A UCS2 character must not be split in the middle; if the length of the User Data Header is odd, the maximum length of the whole TP-UD field is 139 octets.

In the case of compressed GSM 7 bit default alphabet data, 8 bit data or UCS2 the maximum length of the compressed short message within the TP-UD field is 134 (140-6) octets including the Compression Header and Compression Footer, both or either of which may be present (see clause 3.9).

The maximum length of an uncompressed concatenated short message is 39015 (255*153) default alphabet characters, 34170 (255*134) octets or 17085 (255*67) UCS2 characters.

The maximum length of a compressed concatenated message is 34170 (255*134) octets including the Compression Header and Compression Footer (see clause 3.9 and figure 9.2.3.24.1(a) below).

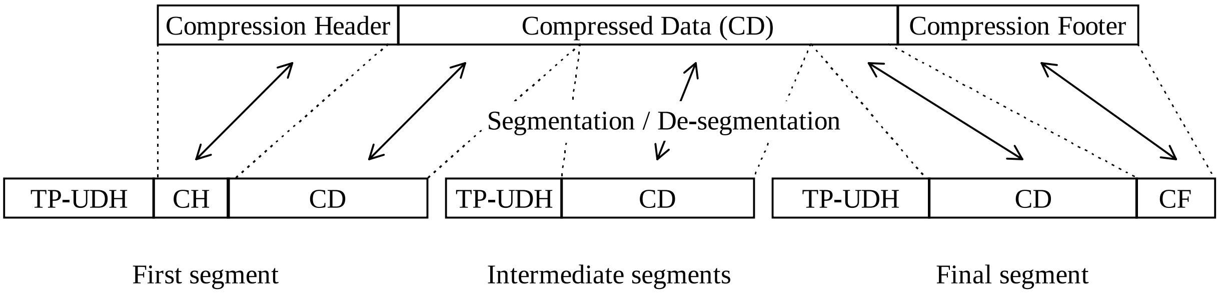


Figure 9.2.3.24.1 (a): Concatenation of a Compressed short message

The Information-Element-Data field contains information set by the application in the SMS-SUBMIT so that the receiving entity is able to re-assemble the short messages in the correct order. Each concatenated short message contains a reference number which together with the originating address and Service Centre address allows the receiving entity to discriminate between concatenated short messages sent from different originating SMEs and/or SCs. In a network which has multiple SCs, it is possible for different segments of a concatenated SM to be sent via different SCs and so it is recommended that the SC address should not be checked by the MS unless the application specifically requires such a check.

The TP elements in the SMS-SUBMIT PDU, apart from TP-MR, TP-SRR, TP-UDL and TP-UD, should remain unchanged for each SM which forms part of a concatenated SM, otherwise this may lead to irrational behaviour. TP-MR must be incremented for every segment of a concatenated message as defined in clause 9.2.3.6. A SC shall handle segments of a concatenated message like any other short message. The relation between segments of a concatenated message is made only at the originator, where the message is segmented, and at the recipient, where the message is reassembled. SMS-COMMANDs identify messages by TP-MR and therefore apply to only one segment of a concatenated message. It is up to the originating SME to issue SMS-COMMANDs for all the required segments of a concatenated message.

The Information-Element-Data octets shall be coded as follows.

Octet 1 Concatenated short message reference number.

This octet shall contain a modulo 256 counter indicating the reference number for a particular concatenated short message. This reference number shall remain constant for every short message which makes up a particular concatenated short message.

Octet 2 Maximum number of short messages in the concatenated short message.

This octet shall contain a value in the range 0 to 255 indicating the total number of short messages within the concatenated short message. The value shall start at 1 and remain constant for every short message which makes up the concatenated short message. If the value is zero then the receiving entity shall ignore the whole Information Element.

Octet 3 Sequence number of the current short message.

This octet shall contain a value in the range 0 to 255 indicating the sequence number of a particular short message within the concatenated short message. The value shall start at 1 and increment by one for every short message sent within the concatenated short message. If the value is zero or the value is greater than the value in octet 2 then the receiving entity shall ignore the whole Information Element.

The IEI and associated IEI length and IEI data shall be present in every segment of the concatenated SM.

9.2.3.24.2 Special SMS Message Indication

There are three levels of "Message Waiting" indication provided within the present document. The first level is to set the Protocol Identifier to "Return Call message", which indicates that a message is waiting and relies on the text of the message to supply the detail. The second level uses the Data Coding Scheme with or without Return Call Message (see 3GPP TS 23.038 [9]) to indicate the type of message waiting and whether there are some messages or no messages. The third level is described here, and provides the maximum detail level for analysis by the mobile, i.e. an indication of the number and type of messages waiting in systems connected to the PLMN. This third level is provided for future flexibility, as it cannot immediately be used without compatibility problems with the earliest Phase mobiles. It is envisaged that this scheme can start to be used once mobiles supporting TP-UDH become widely available.

This information shall be stored by the ME in the Message Waiting Indication Status on the USIM (see 3GPP TS 31.102) when present or otherwise should be stored in the ME. The number of messages shall be stored in Message Waiting Indication Status and an indicator should be shown if the number of messages is non-zero or removed if the number of messages is zero. The ME may also provide some MMI to indicate and access the actual number of messages waiting. Text may be included by the SMS Service Centre for backward compatibility with the earliest Phase mobiles and the Data Coding Scheme may also be used to convey this information in parallel for backward compatibility with "middle" Phase mobiles (which support the use of Data Coding Scheme for Message Waiting Indication but not the use of TP-UDH for Message Waiting Indication).

The information-Element octets shall be coded as follows:

Octet 1 Message Indication type and Storage.

Bit 7 Indicates whether or not the message shall be stored.

Bit 7

0 Discard message after updating indication

1 Store message after updating indication

In the event of a conflict between this setting and the setting of the Data Coding Scheme (see 3GPP TS 23.038 [9]) then the message shall be stored if either the DCS indicates this, or Octet 1 above indicates this.

Bits 6..0 show the message indication type

000 0000	Voice Message Waiting
000 0001	Fax Message Waiting
000 0010	Electronic Mail Message Waiting
000 0011	Other Message Waiting (see 3GPP TS 23.038 [9] for definition of "other")

Other values are reserved for future use.

Octet 2 Message Count.

This octet shall contain a value in the range 0 to 255 indicating the number of messages of the type specified in Octet 1 waiting. The value 255 shall be taken to mean 255 or greater. In the event of a conflict between this setting and the setting of the Data Coding Scheme (see 3GPP TS 23.038 [9]) then the Message Count in the TP-UDH shall override the indication in the TP-DCS.

If more than one type of message is required to be indicated within one SMS message, then further octets must be used, as in the following example:

[00] TP-UDL [1E] (30 decimal septets)

[01] Length of TP-UDH [08]

[02] IEI = Special SMS Message Indication [01]

[03] Length = 02

[04] Octet 1 = Voice Mail, do not store [00]

[05] Octet 2 = 04 Messages

[06] IEI = Special SMS Message Indication [01]

[07] Length = 02

[08] Octet 1 = Fax Mail, Store [81]

[09] Octet 2 = 02 Messages

+ 5 Fill bits

+ 19 seven-bit character message text

The Total number of bits is 210.

In the case where this IEI is to be used in a concatenated SM then the IEI, its associated IEI length and IEI data shall be contained in the first segment of the concatenated SM. The IEI, its associated IEI length and IEI data should also be contained in every subsequent segment of the concatenated SM although this is not mandatory. However, in the case where these elements are not contained in every subsequent segment of the concatenated SM and where an out of sequence segment delivery occurs or where the first segment is not delivered then processing difficulties may arise at the receiving entity which may result in the concatenated SM being totally or partially discarded.

9.2.3.24.3 Application Port Addressing 8 bit address

This facility allows short messages to be routed to one of multiple applications, using a method similar to TCP/UDP ports in a TCP/IP network. An application entity is uniquely identified by the pair of TP-DA/TP-OA and the port address. The port addressing is transparent to the transport, and also useful in Status Reports.

The total length of the IE is 2 octets:

octet 1 Destination port.

This octet contains a number indicating the receiving port, i.e. application, in the receiving device.

octet 2 Originator port.

This octet contains a number indicating the sending port, i.e. application, in the sending device.

The port range is up to 255 using 8 bit addressing space. The Integer value of the port number is presented as in 3GPP TS 23.040 clause 9.1.2.1.

VALUE (port number)	MEANING
0 - 239	Reserved
240 - 255	Available for allocation by applications

A receiving entity shall ignore (i.e. skip over and commence processing at the next information element) any information element where the value of the Information-Element-Data is Reserved or not supported.

In the case where this IE is to be used in a concatenated SM then the IEI, its associated IEI length and IEI data shall be contained in the first segment of the concatenated SM. The IEI, its associated IEI length and IEI data shall also be contained in every subsequent segment of the concatenated SM.

9.2.3.24.4 Application Port Addressing 16 bit address

This facility allows short messages to be routed to one of multiple applications, using a method similar to TCP/UDP ports in a TCP/IP network. An application entity is uniquely identified by the pair of TP-DA/TP-OA and the port address. The port addressing is transparent to the transport, and also useful in Status Reports.

The total length of the IE is 4 octets:

octet 1,2 Destination port.

These octets contain a number indicating the receiving port, i.e. application, in the receiving device.

octet 3,4 Originator port.

These octets contain a number indicating the sending port, i.e. application, in the sending device.

The port range is up to 65535 using 16 bit addressing space. The Integer value of the port number is presented as in 3GPP TS 23.040 clause 9.1.2.1.

VALUE (port number)	MEANING
0 - 15999	As allocated by IANA (http://www.IANA.com/)
16000 - 16999	Available for allocation by applications
17000 - 65535	Reserved

A receiving entity shall ignore (i.e. skip over and commence processing at the next information element) any information element where the value of the Information-Element-Data is Reserved or not supported.

In the case where this IE is to be used in a concatenated SM then the IEI, its associated IEI length and IEI data shall be contained in the first segment of the concatenated SM. The IEI, its associated IEI length and IEI data shall also be contained in every subsequent segment of the concatenated SM.

9.2.3.24.5 SMSC Control Parameters

The facility enables the SMS protocol headers to be expanded using a flexible method. It may be used to control the SMSC, but is also passed transparently to the receiving mobile. The Information Element must be present in every short message affected by it, i.e. in every short message in a concatenated message.

The Information Element data octets shall be coded as follows:

octet 1 Selective Status Report.

This facility is used to control the creation of Status Reports, depending on the error code of the particular message. It is also used by the sending entity to request inclusion of the original UDH into the Status Report. In this case the original UDH must be separated from the rest of the UDH using the Source Indicator. The TP-SRR must be set in order for the Selective Status Report to be enabled. The bits are defined as follows:

bit 0

0 No Status Report for short message transaction completed

1 Status Report for short message transaction completed

bit 1

0 No Status Report for permanent error when SC is not making any more transfer attempts

1 Status Report for permanent error when SC is not making any more transfer attempts

bit 2

0 No Status Report for temporary error when SC is not making any more transfer attempts

1 Status Report for temporary error when SC is not making any more transfer attempts

bit 3

0 No Status Report for temporary error when SC is still trying to transfer SM

1 Status Report for temporary error when SC is still trying to transfer SM

bits 4 and 5

reserved for future use.

bit 6

0 No activation

1 A Status Report generated by this Short Message, due to a permanent error or last temporary error, cancels the SRR of the rest of the Short Messages in a concatenated message. This feature can only be used where a SC is aware of the segmentation of a concatenated SM and is therefore an implementation matter.

bit 7

0 Do not include original UDH into the Status Report

1 Include original UDH into the Status Report

9.2.3.24.6 UDH Source Indicator

The facility is used to separate the UDH of the original message, a UDH created by the SMSC, and a UDH provided by the original receiving entity. The Source Indicator is placed in front of the content inserted by the source. The indicated content (one or more Information-Elements) ends at the next UDH-Source-Indicator, or at the end of the UDH. The Separator is intended to be used especially in Status Reports, but can also be used by the SMSC to add information into Short Message (for example Message waiting). The default content for a UDH in a SMS-DELIVERY is the headers inserted by the sending device, and the default content for a UDH in a SMS-STATUS-REPORT is the headers copied from the SMS-DELIVERY-REPORT.

Values of octet:

01 The following part of the UDH is created by the original sender (valid in case of Status Report)

02 The following part of the UDH is created by the original receiver (valid in case of Status Report)

03 The following part of the UDH is created by the SMSC (can occur in any message or report)

In the case where this IEI is to be used in a concatenated SM then the IEI, its associated IEI length and IEI data shall be contained in the first segment of the concatenated SM. The IEI, its associated IEI length and IEI data should also be contained in every subsequent segment of the concatenated SM although this is not mandatory. However, in the case where these elements are not contained in every subsequent segment of the concatenated SM and where an out of sequence segment delivery occurs or where the first segment is not delivered then processing difficulties may arise at the receiving entity which may result in the concatenated SM being totally or partially discarded.

9.2.3.24.7 (U)SIM Toolkit Security Headers

There are no IEI data values associated with these IEI values and so the associated Length of Information element field is present but set to zero.

These IEI values implicitly define that a Security Header is always present at the start of the TP-User-Data field which immediately follows the TP-User-Data-Header. Details of the Security Header will be found in GSM TS 43.048 [28].

In the case where a concatenated message contains a Security Header then the Security Header will only be present in the first segment of a concatenated message.

In the case where SMS compression is applied to a TP-User-Data field which contains a Security Header then the SMS compression header (3GPP TS 23.042 [26]) shall immediately precede the Security Header.

9.2.3.24.8 Concatenated short messages, 16-bit reference number

This facility is an enhanced variant of the Concatenated Short Message facility (see clause 9.2.3.24.1). The enhancement is a 16-bit reference number, instead of the short 8-bit reference number. The larger reference number reduces the probability that two different concatenated messages are mistakenly sent with identical reference numbers to a receiver. Except for the size of the reference number this facility is identical to the Concatenated Short Message facility (see clause 9.2.3.24.1).

In the case of uncompressed 8-bit data, the maximum length of the short message within the TP-UD field is 133 (140-7) octets.

In the case of uncompressed GSM 7 bit default alphabet data, the maximum length of the short message within the TP-UD field is 151 (160-9) characters.

In the case of 16 bit uncompressed USC2 data, the maximum length of the short message within the TP-UD field is 66 ((140-7)/2) characters. A UCS2 character must not be split in the middle; if the length of the User Data Header is odd, the maximum length of the whole TP-UD field is 139 octets.

In the case of compressed GSM 7 bit default alphabet data, 8 bit data or UCS2 the maximum length of the compressed short message within the TP-UD field is 133 (140-7) octets including the Compression Header and Compression Footer, both or either of which may be present (see clause 3.9).

The relation between compression and concatenation is the same as for Concatenated Short Messages (see clause 9.2.3.24.1).

The Information-Element-Data field contains information set by the application in the SMS-SUBMIT so that the receiving entity is able to re-assemble the short messages in the correct order. Each concatenated short message contains a reference number which together with the originating address and Service Centre address allows the receiving entity to discriminate between concatenated short messages sent from different originating SMEs and/or SCs. In a network which has multiple SCs, it is possible for different segments of a concatenated SM to be sent via different SCs and so it is recommended that the SC address should not be checked by the MS unless the application specifically requires such a check.

The TP elements in the SMS-SUBMIT PDU, apart from TP-MR, TP-UDL and TP-UD, should remain unchanged for each SM which forms part of a concatenated SM, otherwise this may lead to irrational behaviour. TP-MR must be incremented for every segment of a concatenated message as defined in clause 9.2.3.6. A SC shall handle segments of concatenated message like any other short message. The relation between segments of a concatenated message is made at the originator, where the message is segmented, and at the recipient, where the message is reassembled. SMS-COMMANDs identify messages by TP-MR and therefore apply to only one segment of a concatenated message. It is up to the originating SME to issue SMS-COMMANDs for all the required segments of a concatenated message.

The Information-Element-Data octets shall be coded as follows:

Octet 1-2 Concatenated short messages, 16-bit reference number.

This octet shall contain a modulo 65536 counter indicating the reference number for a particular enhanced concatenated short message. This reference number shall remain constant for every short message which makes up a particular enhanced concatenated short message.

Octet 3 Maximum number of short messages in the enhanced concatenated short message.

This octet shall contain a value in the range 0 to 255 indicating the total number of short messages within the concatenated short message. The value shall start at 1 and remain constant for every short message which makes up the enhanced concatenated short message. If the value is zero then the receiving entity shall ignore the whole Information Element.

Octet 4 Sequence number of the current short message.

This octet shall contain a value in the range 0 to 255 indicating the sequence number of a particular short message within the concatenated short message. The value shall start at 1 and increment by one for every short message sent within the concatenated short message. If the value is zero or the value is greater than the value in octet 3 then the receiving entity shall ignore the whole Information Element.

The IEI and associated IEI length and IEI data shall be present in every segment of the concatenated SM.

9.2.3.24.9 Wireless Control Message Protocol

The Wireless Control Message Protocol (WCMP) is part of the WAP suite of protocols; an open standard specified by the WAP Forum Ltd.

The protocol specifies a set of messages that can be used by the receiver to notify the sender if an error occurs. This can be due to routing problems, no application listening at the destination port number, or due to insufficient buffer capacity. The error messages can be used by the sender to avoid retransmitting packets, that can not be properly handled at the receiver. WCMP can also be used for diagnostics and informational purposes. WCMP messages are usually generated by a datagram transport layer or a management entity.

The Information-Element-Data octet(s) shall be coded as follows:

Octet 1-n Protocol Data Unit of WCMP.

This octet(s) shall contain a WCMP protocol data unit.

In the case where this IE is to be used in a concatenated SM then the IEI, its associated IEI length and IEI data shall be contained in the first segment of the concatenated SM. The IEI, its associated IEI length and IEI data shall also be contained in every subsequent segment of the concatenated SM.

9.2.3.24.10 Enhanced Messaging Service

9.2.3.24.10.1 EMS Coding

Enhanced Messaging is based on standard mechanism in GSM SMS messaging. The first mechanism is called **user data header** (TP-UDH), which makes it possible to include binary data in a normal SM prior the text message itself (clause 9.2.3.24). The binary data is in the TP-UD field (message), which means that it steels a part of the 140 bytes. Each object within the SM shall be identified by a IE in the TP-UD Header. The IE will contain a **octet** (refer to clause 9.2.3.24.10.1) that identifies the absolute position of the object within and from the beginning of the SM data. In case of formatting text, an additional octet will give the number of characters for which the formatting applies. Next mechanism that is used is **concatenation,** see clause 9.2.3.24.1. This mechanism permits longer messages than 140 bytes, in fact 255 messages a 140 bytes each can be concatenated to one message up to about 38k bytes.

EMS IEs of the same type may occur more than once in a single message or one segment of a concatenated SM.

9.2.3.24.10.1.1 Text Formatting

The Information-Element-Data octet(s) shall be coded as follows:

Octet 1 Start position of the text formatting. Set to the number of characters after the formatting shall be applied from the beginning of the SM data.

This octet shall be coded as an integer value in the range 0 (beginning of the SM data) to the maximum number of characters included in the SM data of one single SM or one segment of a concatenated SM.

Octet 2 Text formatting length. Gives the number of formatted characters or sets a default text formatting.

This octet shall be coded as an integer value in the range 1 to the maximum number of characters for which the formatting applies in one single SM or one segment of a concatenated SM.

A text formatting length value of 0 indicates that the text format shall be used as a default text format for the current SM. The default text format shall be used for all text in a concatenated SM unless temporarily overridden by a text formatting IE with a non-zero text format length field.

It shall be possible to re-define the default text formatting to be applied to all subsequent text in the current SM by sending a new Text Format IE with text format length zero.

Conflicting overlapping text formatting instructions shall be resolved by applying the formatting instructions in their sequential order.

Octet 3 formatting mode value coded as following:

Octet 3: Bit 7 Bit 6 Bit 5 Bit 4 Bit 3 Bit 2 Bit 1 Bit 0

Bit 1	Bit 0	*Alignment
0	0	Left
0	1	Center
1	0	Right
1	1	Language dependent (default)

*in case formatting text is inserted on the same line as previous non formatting text or with a different mode value, the alignment value shall be set to the same value as the previous formatted predefined object.

Alignment may affect object placement.

Bit 3	Bit 2	Font Size
0	0	Normal (default)
0	1	Large
1	0	Small
1	1	*reserved*

Bit 4	Style **bold**
1	Bold on
0	Bold off

Bit 5	Style *Italic*
1	Italic on
0	Italic off

Bit 6	Style Underlined
1	Underlined on
0	Underlined off

Bit 7	Style ~~Strikethrough~~
1	Strikethrough on
0	Strikethrough off

If bit 4,5,6 and 7 are set to 0, it will mean normal style (default).

Octet 4 Text Colour.

This Octet may be omitted by setting the IED length accordingly.

Bits 0..3 define the Text Foreground Colour

Bits 4..7 define the Text Background Colour

Each colour is defined in a semi octet according to the table below. The actual colours displayed may vary between ME's depending on the display device used.

The colour values defined are simple primary and secondary colours plus four levels of grey. Bright colours have a higher intensity than dark colours.

Nibble Value (msb…lsb)	Colour
0000	Black
0001	Dark Grey
0010	Dark Red
0011	Dark Yellow
0100	Dark Green
0101	Dark Cyan
0110	Dark Blue
0111	Dark Magenta
1000	Grey
1001	White
1010	Bright Red
1011	Bright Yellow
1100	Bright Green
1101	Bright Cyan
1110	Bright Blue
1111	Bright Magenta

9.2.3.24.10.1.2 Predefined Sound

The Information-Element-Data octet(s) shall be coded as follows.

Octet 1 position indicating in the SM data the instant after which the sound shall be played. It will be set to the number of characters from the beginning of the SM data after which the sound shall be played.

This octet shall be coded as an integer value in the range 0 (beginning of the SM data) to the maximum number of characters included in the SM data of one single SM or one segment of a concatenated SM.

Octet 2 sound number. Shall be encoded as a integer value.

9.2.3.24.10.1.3 User Defined Sound

The Information-Element-Data octet(s) shall be coded as follows.

Octet 1 position indicating in the SM data the instant the after which the sound shall be played (refer to clause 9.2.3.24.10.1.2).

Octet 2-n Protocol Data Unit as described in clause 9.2.3.24.10.3.1.

This octet(s) shall contain a User Defined Sound.

9.2.3.24.10.1.4 Predefined Animation

The Information-Element-Data octet(s) shall be coded as follows:

Octet 1 position indicating in the SM data the instant the animation shall be displayed. Set to the number of characters from the beginning of the SM data after which the animation shall be displayed.

This octet shall be coded as an integer value in the range 0 (beginning of the SM data) to the maximum number of characters included in the SM data of one single SM or one segment of a concatenated SM.

Octet 2 animation number. Shall be encoded as an integer value.

9.2.3.24.10.1.5 Large Animation

The Information-Element-Data octet(s) shall be coded as follows:

Octet 1 position indicating the instant the animation shall be displayed in the SM data (refer clause 9.2.3.24.10.1.4).

Octet 2-n Protocol Data Unit as described in clause 9.2.3.24.10.3.3.

This octet(s) shall contain a Large Animation.

9.2.3.24.10.1.6 Small Animation

The Information-Element-Data octet(s) shall be coded as follows:

Octet 1 position indicating the instant the animation shall be displayed in the SM data (refer clause 9.2.3.24.10.1.4).

Octet 2-n Protocol Data Unit as described in clause 9.2.3.24.10.3.3.

This octet(s) shall contain a Small Animation.

9.2.3.24.10.1.7 Large Picture

The Information-Element-Data octet(s) shall be coded as follows:

Octet 1 position indicating in the SM data the instant the picture shall be displayed. Set to the number of characters from the beginning of the SM data after which the picture shall be displayed. This octet shall be coded as an integer value in the range 0 (beginning of the SM data) to the maximum number of characters included in the SM data of one single SM or one segment of a concatenated SM.

Octet 2-n Protocol Data Unit as described in 9.2.3.24.10.3.2.

This octet(s) shall contain a Large Picture.

9.2.3.24.10.1.8 Small Picture

The Information-Element-Data octet(s) shall be coded as follows:

Octet 1 position indicating in the SM data the instant the picture shall be displayed in the SM data (refer clause 9.2.3.24.10.1.7).

Octet 2-n Protocol Data Unit as described in clause 9.2.3.24.10.3.2.

This octet(s) shall contain a Small Picture.

9.2.3.24.10.1.9 Variable Picture

The Information-Element-Data octet(s) shall be coded as follows:

Octet 1 position indicating in the SM data the instant the picture shall be displayed in the SM data (refer clause 9.2.3.24.10.1.7).

Octet 2 Horizontal dimension of the picture.

This octet shall contain the horizontal number of 8 pixels i.e. this value shall be multiplied by 8 to get the whole number of horizontal pixels.

Octet 3 Vertical dimension of the picture.

This octet shall contain the vertical number of pixels.

Octet 4-n Protocol Data Unit as described in clause 9.2.3.24.10.3.2.

This octet(s) shall contain a Variable Picture line by line from top left to bottom right.

The values of the horizontal and vertical dimensions must be chosen properly by the sending entity. If the calculated size of this IE exceeds the limits of a single SM or segment it shall be discarded by the receiving entity.

Examples of EMS coding

All IE values in the TP-UD are hexadecimal values.

9.2.3.24.10.1.10 User Prompt Indicator

With the User Prompt Indicator a sending entity is able to indicate to the receiving entity, that the following object is intended to be handled at the time of reception, e.g. by means of user interaction. The object may be a picture, an animation, a User Defined Sound or a combination of these.

For example the User Prompt Indicator may be used when sending an operators logo to the ME that should be displayed instead of the operators name in standby mode.

When receiving the object the user shall be prompted to accept or discard the object. After this user interaction the SM may be discarded.

The User Prompt Indicator IE shall immediately precede the corresponding object IE(s).

If a User Prompt Indicator IE is not followed by a corresponding object IE it shall be discarded.

The Information-Element-Data octet(s) shall be coded as follows:

Octet 1 Number of corresponding objects.

This octet shall contain the number of corresponding objects as an integer value.

Where Octet 1 indicates that the User Prompt Indicator refers to more than one object, the ME should check the validity of the objects referenced for stitching together. The objects should be considered for stitching if they are either Images (Small, Large, Variable Pictures) or User Defined Sounds, and all of the objects referenced by the User Prompt Indicator IE are of the same type. Animations, Text formatting and pre-defined sound IE's are not suitable for stitching.

User defined sounds may be stitched by concatenating the data contained within each User Defined Sound IE into a single melody object, this may be achieved by ignoring the iMelody header and footer information of the second and subsequent User Defined Sound IE's referenced from the User Prompt Indicator.

Images may be joined along their vertical edges, to form a single "wide" image, the resulting image will have a width equal to the sum of the widths of all the images defined in the User Prompt Indicator.

9.2.3.24.10.1.11 Extended Object

The Extended Object allows an extended code range for format types. The Extended Object may extend across segment boundaries of a concatenated short message. Octets 1 through 7 of the first Extended Object IE shall be contained in a single segment. A single segment may include one or more Extended Object IEs.

If multiple SMs are concatenated and at least one of them contains an Extended Object information element, then concatenation of the SMs shall be done using the 'Concatenated short messages, 16-bit reference number', verses the 'Concatenated short messages, 8-bit reference number' information element. The re-assembly of the Extended Object segments shall be done according to the sequence number of the associated Concatenation IE.

One or more Extended Objects may be compressed using a compression algorithm as indicated in the Compression Control IE (see clause 9.2.3.24.10.1.13).

An SME implementing the Extended Object IE shall be capable of interpreting an uncompressed concatenated message composed of at least **min_eo_msg** short messages which have been received. According to current content provider requirements and handset manufacturer constraints, variable **min_eo_msg** is set to 8.

The first Extended Object IE of an Extended Object contains a reference number, length, control data, type and position. The subsequent Extended Object IEs shall only contain Extended Object data as illustrated in Figure 9.2.24.10.11.

The IE length is variable.

Octet 1 Extended Object reference number.
A modulo 256 counter indicating the reference number for the Extended Object. Two different Extended Objects in a single concatenated message shall have different reference numbers.

Octet 2..3 Extended Object length in number of octets (integer representation) as shown in Figure 9.2.3.24.10.1.11.

Octet 4 Control data.

Bit 0 Object distribution

0 Object may be forwarded
1 Object shall not be forwarded by SMS

Bit 1 User Prompt Indicator

0 Object shall be handled normally
1 Object shall be handled as a User Prompt (see 9.2.3.24.10.1.10)

Bit 2..7 reserved

Any reserved values shall be set to 0.

Octet 5 Extended Object Type.
This octet indicates the format of the Extended Object from the table below.
If the value is reserved or if the associated format is not supported then the receiving entity shall ignore the Extend Object.

Format Type	Format Description
0x00	Predefined sound as defined in annex E.
0x01	iMelody as defined in annex E.
0x02	Black and white bitmap as defined in annex E.
0x03	2-bit greyscale bitmap as defined in annex E.
0x04	6-bit colour bitmap as defined in annex E.
0x05	Predefined animation as defined in annex E.
0x06	Black and white bitmap animation as defined in annex E.
0x07	2-bit greyscale bitmap animation as defined in annex E.
0x08	6-bit colour bitmap animation as defined in annex E.
0x09	vCard as defined in annex E.
0x0A	vCalendar as defined in annex E.
0x0B.. 0xFE	Reserved
0xFF	Data Format Delivery Request as defined in annex E.

Octet 6..7 Extended Object Position (integer representation).
The Extended Object Position indicates the absolute character position within the message text after which the object shall be played or displayed. The absolute character position relates to the entire text within the concatenated message, the first character is numbered character 1.

NOTE: Although this is an absolute value, for concatenated messages, it is suggested the positions used are those that lie within the text of short message segments that have the sequence number equal to or higher than the one that contains the Extended Object IE.

If more than one Extended Object is located at the same position then they may be played or displayed in sequence or simultaneously.

Octet 8..n Extended Object Data.
This sequence of octets is structured as illustrated in the figure below and defined annex E. This figure illustrates the construction of a number of SMs containing a large Extended Object which crosses a SM boundary and is encoded into 2 SM TPDUs. The figure illustrates only the User Data field of the SM (TPDUs). For a description of concatenation of SM refer to Figures 9.2.3.24 (a, b and c)

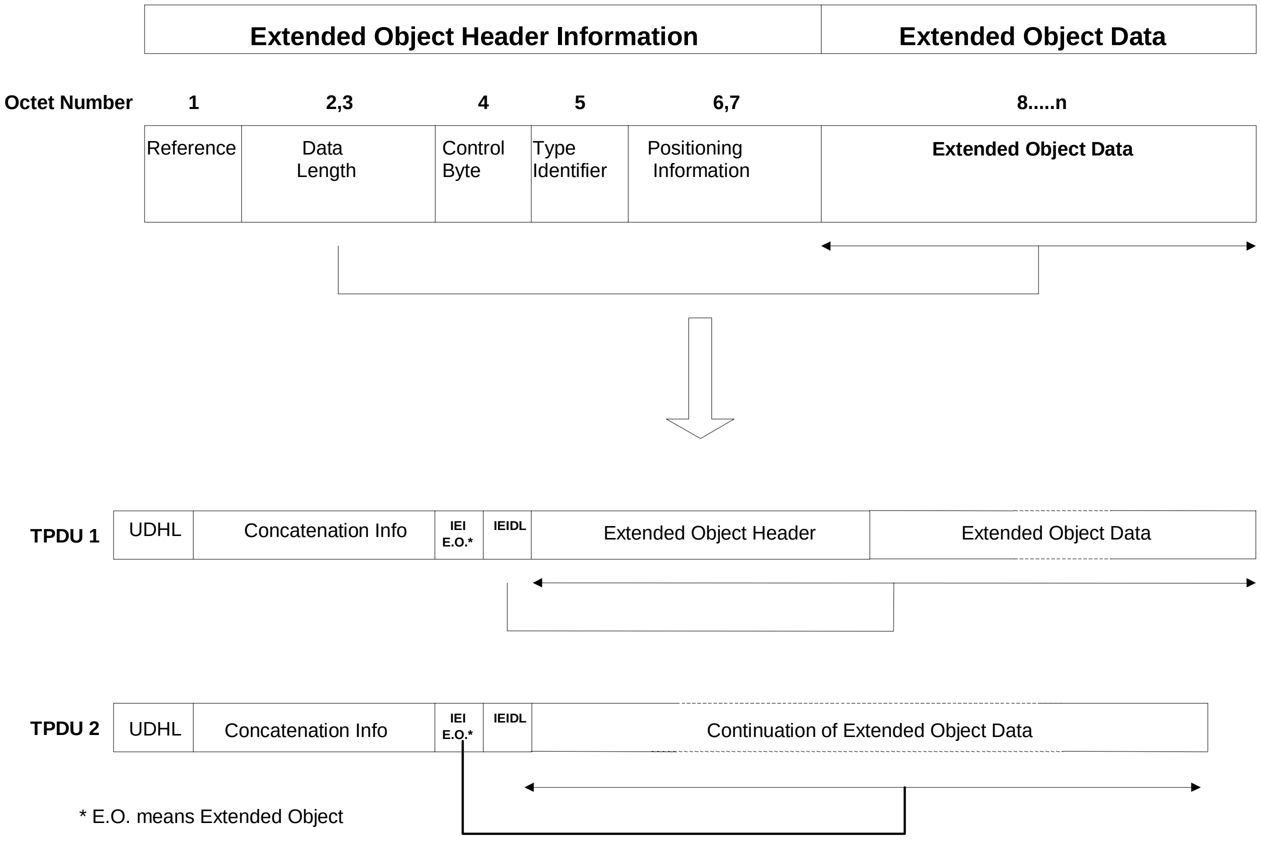


Figure 9.2.3.24.10.1.11

9.2.3.24.10.1.12 Reused Extended Object

This facility is used to reuse an Extended Object in a message which has already been defined in the same message.

Octet 1 Reference number of the Extended Object to be reused.

NOTE: The suggested reference numbers are those of Extended Objects that are contained in short messages that have the sequence number equal to or lower than the one that contains the Reused Extended Object IE.

Octet 2..3 indicates in the concatenated message the absolute character position after which the object shall be played or displayed.

NOTE: Although this is an absolute value, for concatenated messages, the suggested positions that lie within the text of short message segments that have the sequence number equal to or higher than the one that contains the Extended Object IE.

9.2.3.24.10.1.13 Compression Control

This information element is used to indicate a compressed octet sequence. The compression control is only used in association with one or more Extended Objects and/or Reused Extended Objects. The compressed data may extend across sequential short messages within a concatenated short message as illustrated by Figure 9.2.24.10.1.13. The first Compression Control IE of a compressed data sequence contains one octet of Compression Information and a 2-octet length field.

The SME shall support decompression if the Extended Object IE is implemented. An SME implementing the Extending Object IE shall be capable of decompressing a received stream for which the original uncompressed information fits into 1 to **min_eo_msg** messages. An SME may be capable of decompressing a received stream for which the original uncompressed information fits into more than **min_eo_msg** short messages. Variable **min_eo_msg** is defined in clause 9.2.3.24.10.1.11.

The IE length is variable.

Octet 1 Compression information.

Bits 0..3 represent the compression algorithm and bits 4..7 represent compression algorithm specific parameters.

Bit 0..3 Compression algorithm

0000 LZSS Compression according to section 9.2.3.24.10.1.13.1

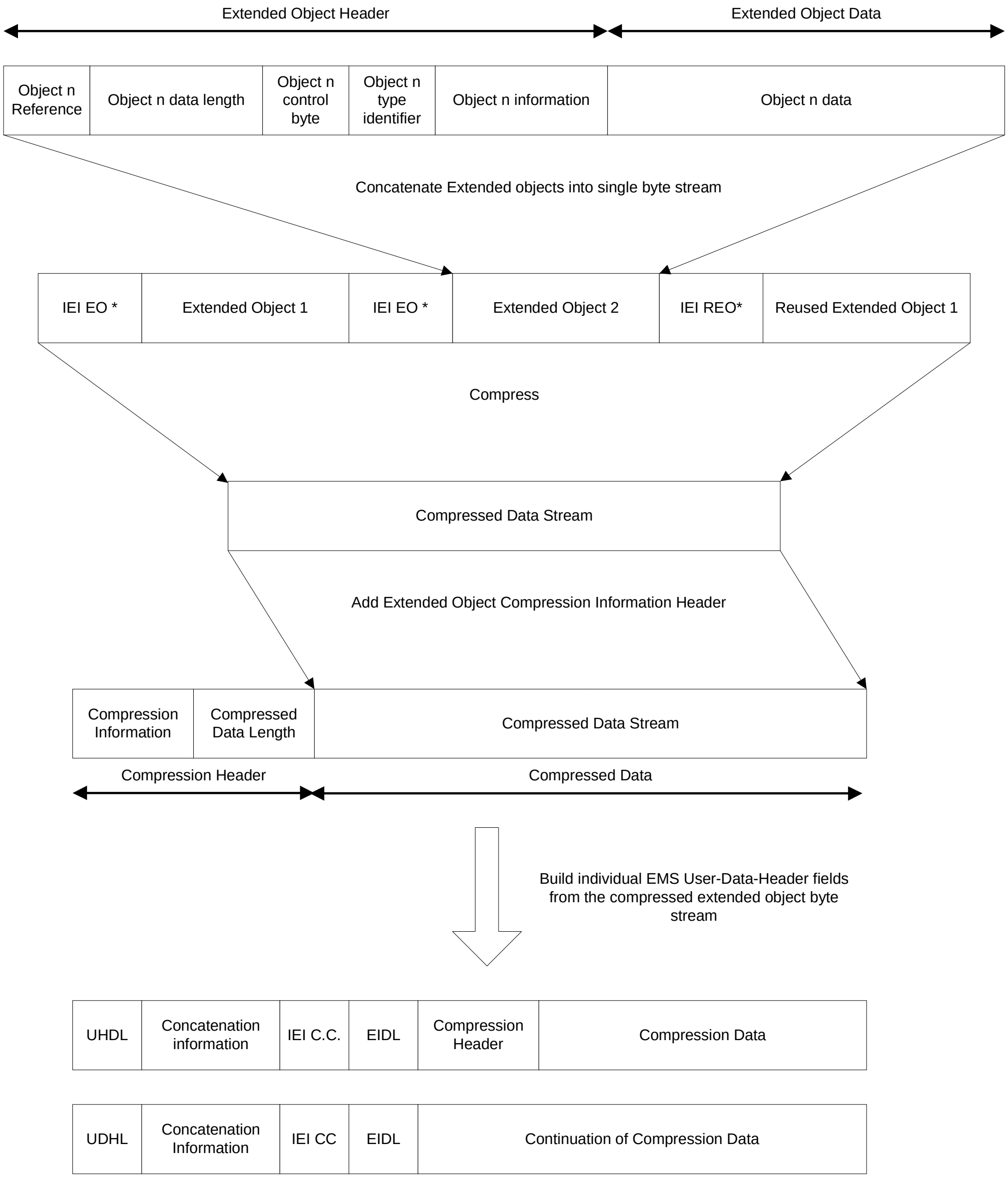
Bit 4..7 Shall be set 0.

0001..1111 reserved for future use; reserved bits shall be transmitted 0.

Bit 4..7 reserved

Octets 2..3 Length of the compressed data in octets (integer representation).
The length indicates the length of the compressed data that may extend across several compression control IEs.

Octets 4..n Compressed data may contain one or more compressed Extended Objects. Figure 9.2.3.24.10.1.13 is an example and illustrates the assembly of a series of SM TPDUs from a sequence of concatenated and compressed extended objects. Each Extended Object is preceded by its IEI (Extended Object or Reused Extended Object). A series of Extended Objects is then compressed into a single buffer and this is split into several SM TPDUs as illustrated.



*E.O Means Extended Object.

*R.E.O means Reused Extended Object.

C.C. means compression.

Figure 9.2.3.24.10.1.13

9.2.3.24.10.1.13.1 LZSS Implementation for EMS extended object compression

LZSS compression uses two tokens to identify either literal strings (byte-sequences) or references to repeated sequences. These tokens (for EMS extended-object compression) are described in this section of the document. A more general introduction to LZSS compression together with an informative example (based upon the tokens described below) is provided in Annex F (informative).

The compressed data stream consists of any combination of literal data blocks and slice descriptor sequences.

The format of the compressed data stream is illustrated as follows: -

<table>
<tr><td colspan="17">Compressed data stream (initial section) …..</td></tr>
<tr><td>1</td><td>2</td><td>3</td><td>4</td><td>5</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td>.</td><td></td><td></td><td></td></tr>
<tr><td colspan="7">Literal data block</td><td colspan="2">Slice descriptor</td><td colspan="4">Literal data block</td><td colspan="2">Slice descriptor</td><td colspan="2">Slice descriptor</td></tr>
</table>

Figure 9.2.3.24.10.1.13.1.a LZSS compressed data format

This diagram represents the structure of a compressed byte stream using LZSS. The stream contains a mixture of literal octets from the input buffer and slice descriptors representing the re-occurrence of an octet sequence together with a length and index for the matching octet sequence. The initial octets of a compressed buffer will always be a sequence of literal octets. The structures of the literal data blocks and slice descriptors are given below.

<table>
<tr><td>Bit 7</td><td>Bit 6</td><td>Bit 5</td><td>Bit 4</td><td>Bit 3</td><td>Bit 2</td><td>Bit 1</td><td>Bit 0</td></tr>
<tr><td>1</td><td colspan="7">Number literal bytes to follow.</td></tr>
</table>

Figure 9.2.3.24.10.1.13.1.b Literal block identifier

When literal octets are written into the compression buffer (for instance during the initial phases of compression) they are preceded by a literal block identifier. The most significant bit (bit 7) of this block shall be set 1. Bits 6-0 indicate the length of the literal block which follows (up to 127 octets). If no match can be found in an octet sequence of greater that 127 octets then 2 (or more) literal blocks shall be written sequentially.

Octet 1								Octet 2							
Bit 15	Bit 14	Bit 13	Bit 12	Bit 11	Bit 10	Bit 9	Bit 8	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
0	Slice Length						Slice Offset								

Figure 9.2.3.24.10.1.13.1.c Slice Descriptor

As can be seen from the above table, the slice descriptor sequence length is two octets, hence only repeating slices of data longer than two octets are extracted. The “slice length” is contained in the descriptor high octet and describes a data slice length of up to 63 octets. The “slice offset index” to the start of the slice is contained in the lower 9 bits and limits the window to 511 octets. The “slice offset index” gives the start position of the source slice measured backwards from the current writing position in the output decoded message data buffer, expressed as a positive number.

9.2.3.24.10.1.13.2 Data Compression

The compressed data output stream is constructed by repeating the following process until the end of the input data buffer is reached.

The input data buffer is scanned, from the current reading position (minus 1) through to a position 511 bytes back from current reading position (the window) looking for the maximum (but limited to 63 octets) length matching data slice contained that matches the data starting at the current reading position (the look ahead buffer).

If no matching data slice, longer than two octets, is found then the input data octet at the current reading position is written to a literal buffer. Both the current reading position in the input data buffer and the current writing position in the output data buffer are incremented by one.

If a matching slice is found then a slice descriptor is written to the output data buffer at the current writing position in the output data buffer and the current writing position is incremented by two. The current reading position in the input data buffer is incremented by the length of the newly found matching data slice.

If the next read octet results in a matching slice being found then the literal buffer is written out. The literal block header, containing a count of the number of literals in the block, is written out first. (If more than 127 literal octets exist in the literal buffer, then it is split into multiple blocks).

The above sequence is repeated until the current reading position reaches the end of the input data buffer.

When encoding (compressing), it is the input data buffer, up to the current reading position, that is used to search for already known matching data slices, as this represents, and is equal to, the reconstructed output data buffer of the decoder at the receiving end.

9.2.3.24.10.1.13.3 Data De-compression

The following sequence is repeated until the end of the input data buffer.

The data octet at the current reading position in the input data buffer is tested for either 0 or 1 in bit 7.

If the bit is set (bit 7 = 1), then the number of literal octets that follow is determined from the lower 7 bits of the header octet (this one).

The literal octet block is written to the output data buffer at the current writing position and both the output data writing position and the input data reading position pointers are incremented by the block size.

If the bit is clear (bit 7 = 0), then the “slice length” and “slice offset index” are extracted from the two octet slice descriptor.

The data slice is copied from within the output data buffer to the end of the output data buffer, where the start of the source slice is at a position “slice offset index” back from the current output data writing position and the destination start position of the slice is the current output buffer writing position. The input data buffer reading position is incremented by two and the output data writing position is incremented by the “slice length”.

9.2.3.24.10.1.13.4 Test Vectors

In order to assist implementers of the compression algorithm described in this specification, a suite of test vectors and ‘help’ information are available in electronic format. The test vectors are supplied on a single diskette attached to this specification.

These test vectors provide checks for most of the commonly expected parameter value variants in this specification and may be updated as the need arises.

In addition Annex F contains an introduction to LZ-type compression algorithms and also has a brief informative example.

9.2.3.24.10.1.14 Object Distribution Indicator

This facility allows a level of control to be requested over the distribution of objects contained within selected information elements in short messages.

If no Object Distribution Indicator is specified for an information element in which an object is received, then that object may be freely distributed.

If a MS provides facilities to modify an object, then the Distribution Attributes (see below) shall be maintained; i.e. an object that is not allowed to be distributed cannot become so after modification.

The use of the Object Distribution Indicator in conjunction with a TE is beyond the scope of the present document.

Where the Object Distribution Indicator is applied to object IE’s that are also addressed by an IE which affects or controls them in some other way (such as User Prompt Indicator IE (see clause 9.2.3.24.10.1.10)), then it shall precede all of the IE’s including the other controlling IE’s.

Octet 1 Number of Information Elements.

This octet specifies the number of information elements from 1-255 for which the Distribution Attributes in the next octet shall apply. The affected objects shall be contained in Information Elements immediately following this IE and may be contained in subsequent short message segments within a concatenated short message.

If the Object Distribution Indicator is applied to the same object IE’s as addressed by an IE which affects or controls them in some other way (such as the User Prompt Indicator IE), then value of this field shall reflect the total number of all the object IE’s and all of the controlling IE’s.

If set to 0 the Distribution Attributes shall apply to all information elements until either the end of the message or another Object Distribution Indicator IE is received.

Octet 2 Distribution Attributes.

Bit 0

0 the associated object(s) may be forwarded

1 the associated object(s) shall not be forwarded by SMS

bit 1..7

reserved for future use.

9.2.3.24.10.1.15 Extended Object Data Request Command

There is no data element associated with this IE. The associated Information Element Length field is present but set to zero.

Upon receiving this IE in an SMS-DELIVER PDU, if an MS supports this request and the corresponding response, it shall respond with an SMS-DELIVER-REPORT PDU containing a Data Format Delivery Request as defined in the Extended Object IE. This SMS-DELIVER PDU may be discarded.

9.2.3.24.10.2.1 Example of Basic text formatting and predefined EMS coding

An example of the basic concept of coding is given as follows:

TP-UDHI=1

SMS User Data Header: UDHL=05, IEI=0A, IEDL=03, IED$_1$=0F, IED$_2$=12, IED$_3$=10

SMS User Data: This is a text with bold option on following with normal text.

Should be displayed as:

This is a text **with bold option on** following with normal text.

It is also possible to add predefined sounds in the message.

Example:

TP-UDHI=1

SMS User Data Header: UDHL=08, IEI=0B, IEDL=02, IED$_1$=09,<sound5>, IEI=0B, IEDL=2, IED$_1$=1C, <sound7>

SMS User Data: This is a message with two different sounds.

The sound nr5 shall be played after the 9th received character ("a") and sound nr7 shall be played after the 28th received character ("e").

9.2.3.24.10.2.2 Example of User defined Objects EMS coding

Example of a message including one small picture is coded as follows:

TP UDHI=1

SMS User Data Header: UDHL=24, IEI=11, IEIDL=22, IED$_1$=08, < ☎ (small picture 32bytes)>

SMS User Data: Hello!<CR><LF><CR><LF>One small picture in here

Should be displayed as:

Hello!

☎

One small picture in here

If the message starts with <CR>, then the "unreadable" data in an old terminal will be overwritten by the text, and the user will not see any strange characters. It is possible to insert the same picture several times in the same message. In that case, the TP-UD header shall contain as many IE as the number of occurrences contained in the SM or one segment of a concatenated message. Using defined elements will normally imply that more than one SM is required and therefore concatenation is required.

9.2.3.24.10.2.3 Concatenation of SMS messages

Concatenated messages are required in most cases required when using several types of EMS elements, since it is only possible to send one large picture/large animation/melody in one single SM. After including either of these elements, there are only 4 (or 9 if no concatenation is used) characters left to the text part, and this is usually too little.

If one or more objects are embedded in one segment of a concatenated message, the IE octet indicating its/their position within the SM data cannot be set to a value that would refer to a position in the next segment(s) so that received segments should be processed before all of them have been received. It means that a formatting text that could not be conveyed in one segment shall be split in as many segments as necessary. In that case, the IE relating to the formatting shall be repeated in all the segments in which it will apply.

Example of a message including 2 Large Pictures, 4 Small animations and 2 User defined Melodies together with some text.

The EMS message: <Large Picture1> <User Defined Melody 1> Hello All, This is a real Enhanced Message <Small Animation 1>. I can send <Small Animation 2> and receive <Small Animation 3> really advanced EMS messages <Animation 4> Isn’t it impressive? /Lars <User Defined Melody2> <Large Picture 2>

This EMS message has to use concatenated messages and the SM will typically contain the following data:

SM	User Data Header	User Data
1	IEI=10 (Large Picture) IED_1=00 (beginning of the SM) <Large Picture 1 (128 bytes)>	[<CR><LF>]
2	IEI=0C (User Defined Sound) IED_1=00 (beginning of the SM) <User Melody 1 (129bytes max)>	Hello
3	IEI=0F (Small Animation) IED_1=24 (36th position) <Small Animation 1 (32 bytes)> IEI=0F (Small Animation) IED_1=2F (47th position) <Small Animation 2 (32 bytes)>	All, This is a real Enhanced Message. I can send and
4	IEI=0F (Small Animation) IED_1=07 (7th position) <Small Animation 3 (32 bytes)> IEI=0F (Small Animation) IED_1=25 (37th position) <Small Animation 4 (32 bytes)>	receive really advanced EMS messages. Isn’t it impressive? /Lars.
5	IEI=0C (User Defined Sound) IED_1=00 (beginning of the SM) <User Melody 1 (128 bytes max)>	[<CR><LF>]
6	IEI=10 (Large Picture) IED_1=00 (beginning of the SM) <Large Picture 2 (128 bytes)>	

9.2.3.24.10.3 EMS Formats

9.2.3.24.10.3.1 Sounds

Predefined Sounds

There are a number of fixed predefined sounds. Each sound nr corresponds to a specific sound according to the table below. The presentations of these sounds are manufacturer specific.

Sound nr	Description
0	Chimes high
1	Chimes low
2	Ding
3	TaDa
4	Notify
5	Drum
6	Claps
7	FanFar
8	Chord high
9	Chord low

User defined sounds

The user defined sounds are coded according to the iMelody format[33]. The maximum length of a sound is 128 bytes.

9.2.3.24.10.3.2 Pictures

Pictures are coded from upper left to lower right and in each byte the most significant bit represent the pixel at the left. The pictures are plain black and white, no colours or grey scales are supported. The bitvalue "0" represents a white pixel and the bitvalue "1" represents a black pixel.

Example 16*16 picture

Byte 1	Byte 2
Byte 3	Byte 4
…	…
…	…
Byte 31	Byte 32

9.2.3.24.10.3.3 Animation

Predefined

There are a number of predefined animations. Each animation nr corresponds to a specific animation according to the table below. The way of displaying the animation is manufacturer specific.

Animation nr	Description
0	I am ironic, flirty
1	I am glad
2	I am sceptic
3	I am sad
4	WOW!
5	I am crying
6	I am winking
7	I am laughing
8	I am indifferent
9	In love/Kissing
10	I am confused
11	Tongue hanging out
12	I am angry
13	Wearing glasses
14	Devil

User Defined

Animations are coded as 4 sequential pictures, with the first picture sent first.

…….

3GPP TSG-T2#16
Sophia-Antipolis, France
11-15 February 2002

T2-020230

CR-Form-v3

CHANGE REQUEST

⌘	**23.040**	**CR**	**047**	⌘	rev	**-**	⌘	Current version:	**5.2.0**	⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network □ Core Network □

Title: ⌘	Alternate Reply Address Element			
Source: ⌘	T2			
Work item code: ⌘	TEI5		***Date:*** ⌘	30 January 2002
Category: ⌘	**B**		***Release:*** ⌘	REL-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘	There is no easy and standardized way to share a single SMSC access to host multiple two-way SMS/EMS applications and in some cases, it is desirable to set a return address different of the one used to send the SMS (all SMSC don't allow to freely specify Originating Address). It also provides a mechanism to reply with an SMS-PP to a message initially send as SMS-CB.
Summary of change: ⌘	Introduce a new information element in EMS
Consequences if not approved: ⌘	Deployment of two-way SMS/EMS VAS will still be challenged, due to the costs of using dedicated accesses for every single application.

Clauses affected: ⌘	9.2.3.24, 9.2.3.24.10.11.15 (added)	
Other specs affected: ⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘
Other comments: ⌘		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

9.2.3.24 TP-User Data (TP-UD)

The length of the TP-User-Data field is defined in the PDU's of the SM-TL (see clause 9.2.2).

The TP-User-Data field may comprise just the short message itself or a Header in addition to the short message depending upon the setting of TP-UDHI.

Where the TP-UDHI value is set to 0 the TP-User-Data field comprises the short message only, where the user data can be 7 bit (default alphabet) data, 8 bit data, or 16 bit (UCS2 [24]) data.

Where the TP-UDHI value is set to 1 the first octets of the TP-User-Data field contains a Header in the following order starting at the first octet of the TP-User-Data field.

Irrespective of whether any part of the User Data Header is ignored or discarded, the MS shall always store the entire TPDU exactly as received.

FIELD	LENGTH
Length of User Data Header	1 octet
Information-Element-Identifier "A"	1 octet
Length of Information-Element "A"	1 octet
Information-Element "A" Data	0 to "n" octets
Information-Element-Identifier "B"	1 octet
Length of Information-Element "B"	1 octet
Information-Element "B" Data	0 to "n" octets
Information-Element-Identifier "X"	1 octet
Length of Information-Element "X"	1 octet
Information-Element "X" Data	0 to "n" octets

The diagram below shows the layout of the TP-User-Data-Length and the TP-User-Data for uncompressed GSM 7 bit default alphabet data. The UDHL field is the first octet of the TP-User-Data content of the Short Message.

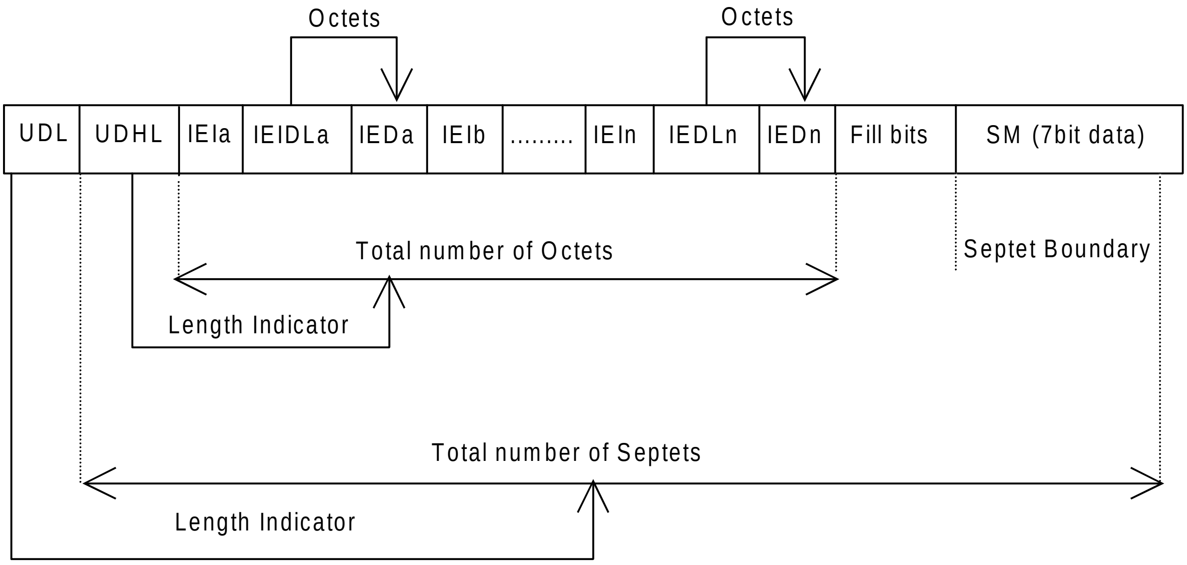


Figure 9.2.3.24 (a)

The diagram below shows the layout of the TP-User-Data-Length and the TP-User-Data for uncompressed 8 bit data or uncompressed UCS2 data. The UDHL field is the first octet of the TP-User-Data content of the Short Message.

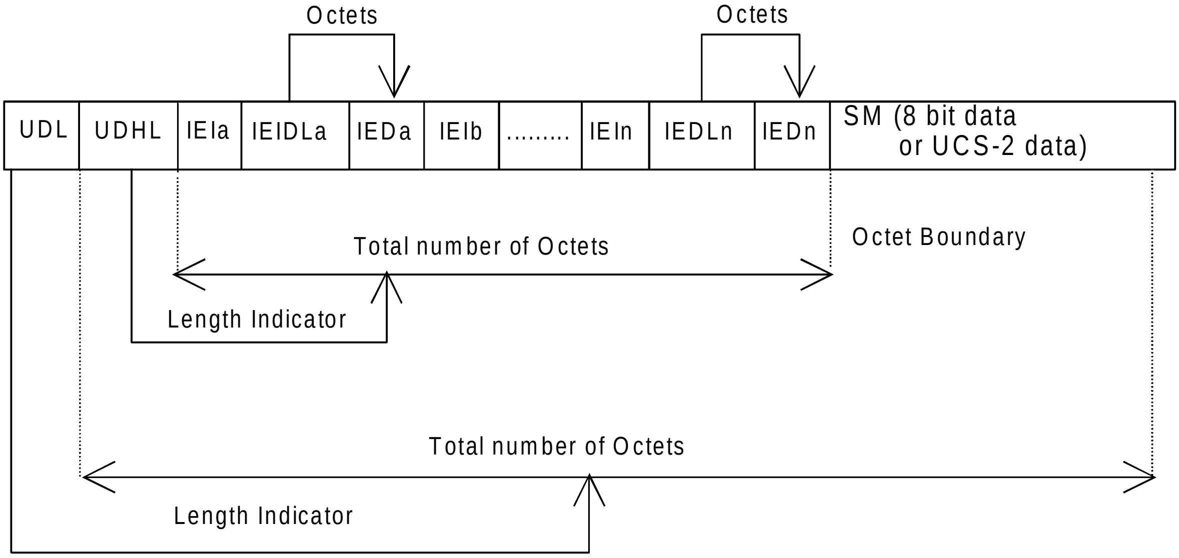


Figure 9.2.3.24 (b)

The diagram below shows the layout of the TP-User-Data-Length and the TP-User-Data for compressed GSM 7 bit default alphabet data, compressed 8 bit data or compressed UCS2 data. The UDHL field is the first octet of the TP-User-Data content of the Short Message.

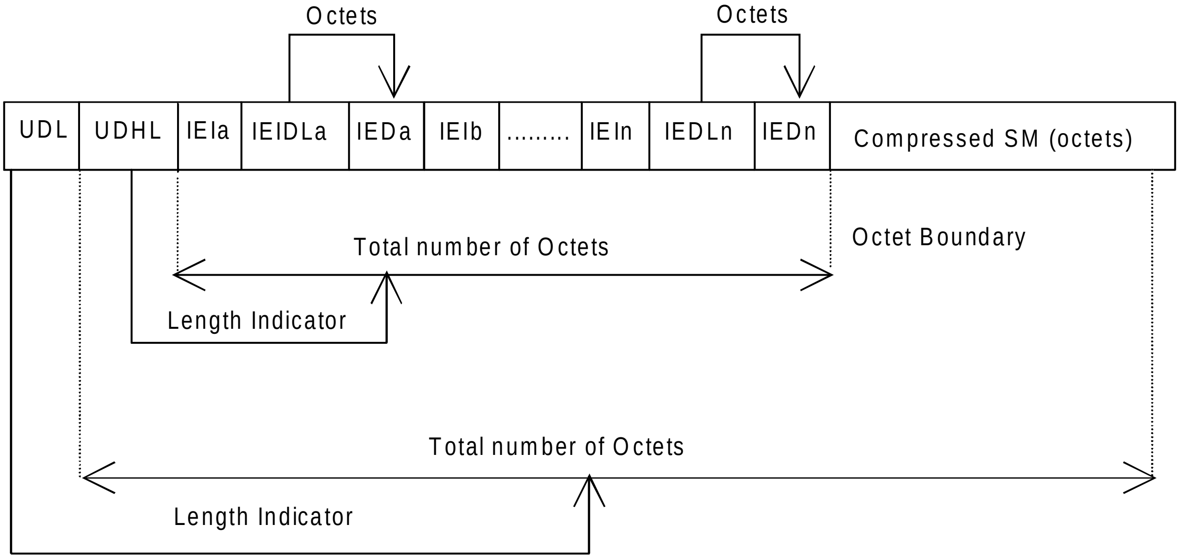


Figure 9.2.3.24 (c)

The definition of the TP-User-Data-Length field which immediately precedes the "Length of User Data Header" is unchanged and shall therefore be the total length of the TP-User-Data field including the Header, if present. (see 9.2.3.16).

The "Length-of-Information-Element" fields shall be the integer representation of the number of octets within its associated "Information-Element-Data" field which follows and shall not include itself in its count value.

The "Length-of-User-Data-Header" field shall be the integer representation of the number of octets within the "User-Data-Header" information fields which follow and shall not include itself in its count or any fill bits which may be present (see text below).

Information Elements may appear in any order and need not follow the order used in the present document. Information Elements are classified into 3 categories as described below.

- SMS Control – identifies those IEIs which have the capability of dictating SMS functionality.
- EMS Control – identifies those IEIs which manage EMS Content IEIs.

- EMS Content – identifies those IEIs containing data of a unique media format.

It is permissible for certain IEs to be repeated within a short message, or within a concatenated message. There is no restriction on the repeatability of IEs in the EMS Content classification. The repeatability of SMS Control and EMS Control IEs is determined on an individual basis. See the IE table below for the repeatability of each IE.

In the event that IEs determined as not repeatable are duplicated, the last occurrence of the IE shall be used. In the event that two or more IEs occur which have mutually exclusive meanings (e.g. an 8bit port address and a 16bit port address), then the last occurring IE shall be used.

If the length of the User Data Header is such that there are too few or too many octets in the final Information Element then the whole User Data Header shall be ignored.

If any reserved values are received within the content of any Information Element then that part of the Information Element shall be ignored.

The Information Element Identifier octet shall be coded as follows:

VALUE (hex)	MEANING	Classification	Repeatability
00	Concatenated short messages, 8-bit reference number	SMS Control	No
01	Special SMS Message Indication	SMS Control	Yes
02	Reserved	N/A	N/A
03	Value not used to avoid misinterpretation as <LF> character	N/A	N/A
04	Application port addressing scheme, 8 bit address	SMS Control	No
05	Application port addressing scheme, 16 bit address	SMS Control	No
06	SMSC Control Parameters	SMS Control	No
07	UDH Source Indicator	SMS Control	Yes
08	Concatenated short message, 16-bit reference number	SMS Control	No
09	Wireless Control Message Protocol	SMS Control	Note 3
0A	Text Formatting	EMS Control	Yes
0B	Predefined Sound	EMS Content	Yes
0C	User Defined Sound (iMelody max 128 bytes)	EMS Content	Yes
0D	Predefined Animation	EMS Content	Yes
0E	Large Animation (16*16 times 4 = 32*4 =128 bytes)	EMS Content	Yes
0F	Small Animation (8*8 times 4 = 8*4 =32 bytes)	EMS Content	Yes
10	Large Picture (32*32 = 128 bytes)	EMS Content	Yes
11	Small Picture (16*16 = 32 bytes)	EMS Content	Yes
12	Variable Picture	EMS Content	Yes
13	User prompt indicator	EMS Control	Yes
14	Extended Object	EMS Content	Yes
15	Reused Extended Object	EMS Control	Yes
16	Compression Control	EMS Control	No
17	Object Distribution Indicator	EMS Control	Yes
18-1F	Reserved for future EMS features (see subclause 3.10)	N/A	N/A
20	RFC 822 E-Mail Header	SMS Control	No
21	Hyperlink format element	SMS Control	Yes
22	Reply Address Element	SMS Control	No
~~22~~ 23 – 6F	Reserved for future use	N/A	N/A
70 – 7F	(U)SIM Toolkit Security Headers	SMS Control	Note 1
80 – 9F	SME to SME specific use	SMS Control	Note 2
A0 – BF	Reserved for future use	N/A	N/A
C0 – DF	SC specific use	SMS Control	Note 2
E0 – FF	Reserved for future use	N/A	N/A

Note 1: The functionality of these IEIs is defined in 3GPP TSG 23.048 [28], and therefore, the repeatability is not within the scope of this document and will not be determined here.

Note 2: The functionality of these IEIs is used in a proprietary fashion by different SMSC vendors, and therefore, are not within the scope of this technical specification.

Note 3: The functionality of these IEIs is defined by the WAP Forum and therefore the repeatability is not within the scope of this document and will not be determined here.

A receiving entity shall ignore (i.e. skip over and commence processing at the next information element) any information element where the IEI is Reserved or not supported. The receiving entity calculates the start of the next information element by looking at the length of the current information element and skipping that number of octets.

The SM itself may be coded as 7, 8 or 16 bit data.

If 7 bit data is used and the TP-UD-Header does not finish on a septet boundary then fill bits are inserted after the last Information Element Data octet up to the next septet boundary so that there is an integral number of septets for the entire TP-UD header. This is to ensure that the SM itself starts on an septet boundary so that an earlier Phase mobile shall be capable of displaying the SM itself although the TP-UD Header in the TP-UD field may not be understood.

It is optional to make the first character of the SM itself a Carriage Return character encoded according to the default 7 bit alphabet so that earlier Phase mobiles, which do not understand the TP-UD-Header, shall over-write the displayed TP-UD-Header with the SM itself.

If 16 bit (USC2) data is used then padding octets are not necessary. The SM itself shall start on an octet boundary.

If 8 bit data is used then padding is not necessary. An earlier Phase mobile shall be able to display the SM itself although the TP-UD header may not be understood.

It is also possible for mobiles not wishing to support the TP-UD header to check the value of the TP-UDHI bit in the SMS-Deliver PDU and the first octet of the TP-UD field and skip to the start of the SM and ignore the TP-UD header.

...

9.2.3.24.10.1.15 Reply Address Element

Only one alternate Reply Address Element can be integrated in a message

Octet 1..n Alternate Reply Address encoded as specified for address fields in clause 9.1.2.5

When this IE is received in a message, replies to this message should take place by default using the address specified in this IE instead of the regular message TP-OA.

> Note: Despite the fact that MMI aspects of the ME are out of the scope of this specification, it must be mentioned that this mechanism might open the door to potential abuse. It is desirable that the user is made aware in some way that the reply address of the incoming message is different from the originator's one, and that the user is presented with the original TP-OA address to identify the sender of the SM .

3GPP TSG-T2 #16
Sophia-Antipolis, France
11-15 February 2002

T2-020304

CR-Form-v3

CHANGE REQUEST

⌘ **23.040 CR 046** ⌘ rev **1** ⌘ Current version: **5.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network □ Core Network □

Title:	⌘	Subaddressing scheme for SMS		
Source:	⌘	T2		
Work item code:	⌘	TEI5	***Date:*** ⌘	13 February 2001
Category:	⌘	**B**	***Release:*** ⌘	REL-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change:	⌘	There is no easy and standardized way to share a single SMSC access to host multiple two-way SMS/EMS applications.
Summary of change:	⌘	Introduce # as a separator, and remaining digits as a subaddress to be conveyed by the SMSC from SME to SME
Consequences if not approved:	⌘	Deployment of two-way SMS/EMS VAS will still be challenged, due to the costs of using dedicated accesses for every single application.

Clauses affected:	⌘	6.2.1 (added), 9.2.3.7, 9.2.3.8
Other specs affected:	⌘	□ Other core specifications ⌘ □ Test specifications □ O&M Specifications
Other comments:	⌘	

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

6 Service Centre functionality

In the present document, only the SC functionality related to the short message service between the SC and the MS is specified.

6.1 Service Centre capabilities

The SC should be capable of:

- submitting a short message to an MS, retaining the responsibility of the message until

  1) the report has been received; or

  2) the Validity-Period expires.

- receiving a report from the PLMN;

- receiving a short message from an MS;

- returning a report to the PLMN for a previously received short message.

6.2 SC functional requirements

The detailed functionality of the SC is outside the scope of the present document, and is for the SC operator to define. However, the following functional requirements are mandatory for all SCs in order to support the SM-TP (see clause 9) towards the PLMN:

1) To identify each SMS-DELIVER sent to an MS in a unique way, a time stamp value is included in the field TP-Service-Centre-Time-Stamp, TP-SCTS, of the SMS-DELIVER. The time stamp gives the time when the message arrived at the SC with the accuracy of a second. If two or more messages to the same MS arrive at the SC within one second, the SC shall modify the time stamp of those messages in such a way that:

   a) all messages to the MS contain different time stamps;

   b) the modification of the time stamps is kept to a minimum.

2) The SC is only allowed to have one outstanding SMS-DELIVER (i.e. a message for which a report has not been received) to a specific MS at a given time.

3) The SC shall be able to initiate overwriting of short messages previously received by the SC if requested by the same originating address (MS or any other source) by use of the same message type.

6.2.1 Subaddressing support

Support for subaddressing is an optional functional requirement for an SC. If it is supported, subaddressing information shall be conveyed from SME to SME according the following rules:

- A SME may send a SM with ‘*’s or ‘#’s included in the TP-DA field. The first ‘#’ encountered in TP-DA indicates where the address for SC routing purposes is terminated. Additional ‘*’s or ‘#’s can be present in the following digits, and all these digits including the first ‘#’ are subaddress digits.

- When the SC receives a SM to convey with such a subaddress information, it should deliver the SM to the destination SME with the same subaddress digits copied in the TP-OA field.

This subaddressing mechanism does not apply when the TON is alphanumeric

Example:

SME with number 987654321 sends a SM with TP-DA = 1234#56#789*

SME with number 1234 will receive the SM with TP-OA = 987654321#56#789*

7 MS functionality

…

9.2.3.7 TP-Originating-Address (TP-OA)

The TP-Originating-Address field is formatted according to the formatting rules of address fields.

The first '#' encountered in TP-OA indicates where the address for SMSC routing purposes is terminated. Additional '*'s or '#'s can be present in the following digits, and all these digits including the first '#' are subaddress digits.

9.2.3.8 TP-Destination-Address (TP-DA)

The TP-Destination-Address field is formatted according to the formatting rules of address fields.

The first '#' encountered in TP-DA indicates where the address for SMSC routing purposes is terminated. Additional '*'s or '#'s can be present in the following digits, and all these digits including the first '#' are subaddress digits.

9.2.3.9 TP-Protocol-Identifier (TP-PID)

…

TSG-T WG 2 meeting #16 **TSG T2- 020227**
Sophia Antipolis, France 11-15 February 2002 **Agenda Item:**

CR-Form-v3

CHANGE REQUEST

⌘	**23.040**	**CR**	**045**	⌘	rev	**-**	⌘	Current version:	**5.2.0** ⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘	(U)SIM	□	ME/UE	☒	Radio Access Network	□	Core Network	☒

Title:	⌘	MO-SMS duplicate message response		
Source:	⌘	T2		
Work item code:	⌘	TEI5	***Date:*** ⌘	11-15 Feb. 2002
Category:	⌘	**A**	***Release:*** ⌘	REL-5
		Use one of the following categories: ***F*** *(essential correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(Addition of feature),* ***C*** *(Functional modification of feature)* ***D*** *(Editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*

Reason for change:	⌘	Editorial correction and clarification for an inconsistency in section 9.2.3.6 and 9.2.3.25. (This inconsistency has made by an editorial error in T2-991065.)
Summary of change:	⌘	This CR corrects the editorial error that has made an inconsistency and clarifies as follows. (1) The SC response to a duplicate message, where TP-RD=1, should be an RP-ERROR, in which case the RP-ERROR shall include a SMS-SUBMIT-REPORT with TP-FCS indicating “SM Rejected – Duplicate SM” (2) In some cases, for backward compatibility with earlier phases and versions of this specification, the SC may be configured to respond with an RP-ACK.
Consequences if not approved:	⌘	*Editorial error in subclause 9.2.3.6 *Inconsistency in TS23.040

Clauses affected:	⌘	9.2.3.6 and 9.2.3.25		
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

9.2.3.6 TP-Message-Reference (TP-MR)

The TP-Message-Reference field gives an integer representation of a reference number of the SMS-SUBMIT or SMS-COMMAND submitted to the SC by the MS. The MS increments TP-Message-Reference by 1 for each SMS-SUBMIT or SMS-COMMAND being submitted. The value to be used for each SMS-SUBMIT is obtained by reading the Last-Used-TP-MR value from the SMS Status data field in the (U)SIM (see GSM TS 11.11 [16] and 3G TS 31.102 [30]) and incrementing this value by 1. After each SMS-SUBMIT has been submitted to the network, the Last-Used-TP-MR value in the (U)SIM is updated with the TP-MR that was used in the SMS-SUBMIT operation. The reference number may possess values in the range 0 to 255. The value in the TP-MR assigned by the MS is the same value which is received at the SC.

In the case where no response or an ~~RP-Error~~RP-ERROR with an appropriate cause value (see 3GPP TS 24.011 [13]) is received in response to an SMS-SUBMIT or SMS-COMMAND, then the MS shall automatically repeat the SMS-SUBMIT or SMS-COMMAND but must use the same TP-MR value and set the TP-RD bit to 1 (see 9.2.3.25). The number of times the MS automatically repeats the SMS-SUBMIT or SMS-COMMAND shall be in the range 1 to 3 but the precise number is an implementation matter. The automatic repeat mechanism should be capable of being disabled through MMI.

If all automatic attempts fail (or in the case of no automatic attempts the first attempt fails), the user shall be informed. The failed message shall be stored in the mobile in such a way that the user can request a retransmission using the same TP-MR value, without the need to re-enter any information. Such storage need only be provided for a single failed message, i.e. the one most recently attempted.

The SC should discard an SMS-SUBMIT or SMS-COMMAND which has the TP-RD bit set to a 1 and which has the same TP-MR value as the previous SMS-SUBMIT or SMS-COMMAND received from the same originating address. ~~The SC shall send an RP-ACK in response to~~ In the case of a ~~the~~ discarded SMS-SUBMIT or SMS-COMMAND, the SC should respond with an RP-ERROR, in which case the RP-ERROR shall include a SMS-SUBMIT-REPORT with TP-FCS indicating “SM Rejected – Duplicate SM”. In some cases, for backward compatibility with earlier phases and versions of this specification, the SC may be configured to respond with an RP-ACK.

The SMS-STATUS-REPORT also contains a TP-Message-Reference field. The value sent to the MS shall be the same as the TP-Message-Reference value generated by the MS in the earlier SMS-SUBMIT or SMS-COMMAND to which the status report relates.

9.2.3.25 TP-Reject-Duplicates (TP-RD)

The TP-Reject-Duplicates is a 1 bit field located within bit 2 of the first octet of SMS-SUBMIT and has the following values.

Bit no. 2: 0 Instruct the SC to accept an SMS-SUBMIT for an SM still held in the SC which has the same TP-MR and the same TP-DA as a previously submitted SM from the same OA.

1 Instruct the SC to reject an SMS-SUBMIT for an SM still held in the SC which has the same TP-MR and the same TP-DA as the previously submitted SM from the same OA. In this case ~~an appropriate TP-FCS value shall be returned in the SMS-SUBMIT-REPORT~~the response returned by the SC is as specified in 9.2.3.6.

TSG-T WG 2 meeting #16
Sophia Antipolis, France 11-15 February 2002

TSG T2- 020226
Agenda Item:

CR-Form-v3

CHANGE REQUEST

⌘ **23.040 CR 044** ⌘ rev **-** ⌘ Current version: **4.5.0** ⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network □ Core Network ☒

Title: ⌘	MO-SMS duplicate message response		
Source: ⌘	T2		
Work item code: ⌘	TEI	***Date:*** ⌘	11-15 Feb. 2002
Category: ⌘	**A** *Use one of the following categories:* ***F*** *(essential correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(Addition of feature),* ***C*** *(Functional modification of feature)* ***D*** *(Editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.	***Release:*** ⌘	REL-4 *Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*

Reason for change: ⌘	Editorial correction and clarification for an inconsistency in section 9.2.3.6 and 9.2.3.25. (This inconsistency has made by an editorial error in T2-991065.)
Summary of change: ⌘	This CR corrects the editorial error that has made an inconsistency and clarifies as follows. (1) The SC response to a duplicate message, where TP-RD=1, should be an RP-ERROR, in which case the RP-ERROR shall include a SMS-SUBMIT-REPORT with TP-FCS indicating "SM Rejected – Duplicate SM" (2) In some cases, for backward compatibility with earlier phases and versions of this specification, the SC may be configured to respond with an RP-ACK.
Consequences if not approved: ⌘	*Editorial error in subclause 9.2.3.6 *Inconsistency in TS23.040

Clauses affected: ⌘	9.2.3.6 and 9.2.3.25	
Other specs affected: ⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘
Other comments: ⌘		

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

9.2.3.6 TP-Message-Reference (TP-MR)

The TP-Message-Reference field gives an integer representation of a reference number of the SMS-SUBMIT or SMS-COMMAND submitted to the SC by the MS. The MS increments TP-Message-Reference by 1 for each SMS-SUBMIT or SMS-COMMAND being submitted. The value to be used for each SMS-SUBMIT is obtained by reading the Last-Used-TP-MR value from the SMS Status data field in the (U)SIM (see GSM TS 11.11 [16] and 3G TS 31.102 [30]) and incrementing this value by 1. After each SMS-SUBMIT has been submitted to the network, the Last-Used-TP-MR value in the (U)SIM is updated with the TP-MR that was used in the SMS-SUBMIT operation. The reference number may possess values in the range 0 to 255. The value in the TP-MR assigned by the MS is the same value which is received at the SC.

In the case where no response or an ~~RP-Error~~RP-ERROR with an appropriate cause value (see 3GPP TS 24.011 [13]) is received in response to an SMS-SUBMIT or SMS-COMMAND, then the MS shall automatically repeat the SMS-SUBMIT or SMS-COMMAND but must use the same TP-MR value and set the TP-RD bit to 1 (see 9.2.3.25). The number of times the MS automatically repeats the SMS-SUBMIT or SMS-COMMAND shall be in the range 1 to 3 but the precise number is an implementation matter. The automatic repeat mechanism should be capable of being disabled through MMI.

If all automatic attempts fail (or in the case of no automatic attempts the first attempt fails), the user shall be informed. The failed message shall be stored in the mobile in such a way that the user can request a retransmission using the same TP-MR value, without the need to re-enter any information. Such storage need only be provided for a single failed message, i.e. the one most recently attempted.

The SC should discard an SMS-SUBMIT or SMS-COMMAND which has the TP-RD bit set to a 1 and which has the same TP-MR value as the previous SMS-SUBMIT or SMS-COMMAND received from the same originating address. ~~The SC shall send an RP-ACK in response to~~ In the case of a ~~the~~ discarded SMS-SUBMIT or SMS-COMMAND, the SC should respond with an RP-ERROR, in which case the RP-ERROR shall include a SMS-SUBMIT-REPORT with TP-FCS indicating “SM Rejected – Duplicate SM”. In some cases, for backward compatibility with earlier phases and versions of this specification, the SC may be configured to respond with an RP-ACK.

The SMS-STATUS-REPORT also contains a TP-Message-Reference field. The value sent to the MS shall be the same as the TP-Message-Reference value generated by the MS in the earlier SMS-SUBMIT or SMS-COMMAND to which the status report relates.

9.2.3.25 TP-Reject-Duplicates (TP-RD)

The TP-Reject-Duplicates is a 1 bit field located within bit 2 of the first octet of SMS-SUBMIT and has the following values.

Bit no. 2: 0 Instruct the SC to accept an SMS-SUBMIT for an SM still held in the SC which has the same TP-MR and the same TP-DA as a previously submitted SM from the same OA.

1 Instruct the SC to reject an SMS-SUBMIT for an SM still held in the SC which has the same TP-MR and the same TP-DA as the previously submitted SM from the same OA. In this case ~~an appropriate TP-FCS value shall be returned in the SMS-SUBMIT-REPORT~~the response returned by the SC is as specified in 9.2.3.6.

TSG-T WG 2 meeting #16 **TSG T2- 020225**
Sophia Antipolis, France 11-15 February 2002 **Agenda Item:**

CR-Form-v3

CHANGE REQUEST

⌘	**23.040**	**CR**	**043**	⌘	rev	**-**	⌘	Current version:	**3.7.0**	⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network □ Core Network ☒

Title:	⌘	MO-SMS duplicate message response		
Source:	⌘	T2		
Work item code:	⌘	TEI	***Date:*** ⌘	11-15 Feb. 2002
Category:	⌘	**F**	***Release:*** ⌘	R99
		Use one of the following categories: ***F*** *(essential correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(Addition of feature),* ***C*** *(Functional modification of feature)* ***D*** *(Editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*

Reason for change:	⌘	Editorial correction and clarification for an inconsistency in section 9.2.3.6 and 9.2.3.25. (This inconsistency has made by an editorial error in T2-991065.)
Summary of change:	⌘	This CR corrects the editorial error that has made an inconsistency and clarifies as follows. (1) The SC response to a duplicate message, where TP-RD=1, should be an RP-ERROR, in which case the RP-ERROR shall include a SMS-SUBMIT-REPORT with TP-FCS indicating "SM Rejected – Duplicate SM" (2) In some cases, for backward compatibility with earlier phases and versions of this specification, the SC may be configured to respond with an RP-ACK.
Consequences if not approved:	⌘	*Editorial error in subclause 9.2.3.6 *Inconsistency in TS23.040

Clauses affected:	⌘	9.2.3.6 and 9.2.3.25		
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:
Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

9.2.3.6 TP-Message-Reference (TP-MR)

The TP-Message-Reference field gives an integer representation of a reference number of the SMS-SUBMIT or SMS-COMMAND submitted to the SC by the MS. The MS increments TP-Message-Reference by 1 for each SMS-SUBMIT or SMS-COMMAND being submitted. The value to be used for each SMS-SUBMIT is obtained by reading the Last-Used-TP-MR value from the SMS Status data field in the (U)SIM (see GSM TS 11.11 [16] and 3G TS 31.102 [30]) and incrementing this value by 1. After each SMS-SUBMIT has been submitted to the network, the Last-Used-TP-MR value in the (U)SIM is updated with the TP-MR that was used in the SMS-SUBMIT operation. The reference number may possess values in the range 0 to 255. The value in the TP-MR assigned by the MS is the same value which is received at the SC.

In the case where no response or an ~~RP-Error~~RP-ERROR with an appropriate cause value (see 3GPP TS 24.011 [13]) is received in response to an SMS-SUBMIT or SMS-COMMAND, then the MS shall automatically repeat the SMS-SUBMIT or SMS-COMMAND but must use the same TP-MR value and set the TP-RD bit to 1 (see 9.2.3.25). The number of times the MS automatically repeats the SMS-SUBMIT or SMS-COMMAND shall be in the range 1 to 3 but the precise number is an implementation matter. The automatic repeat mechanism should be capable of being disabled through MMI.

If all automatic attempts fail (or in the case of no automatic attempts the first attempt fails), the user shall be informed. The failed message shall be stored in the mobile in such a way that the user can request a retransmission using the same TP-MR value, without the need to re-enter any information. Such storage need only be provided for a single failed message, i.e. the one most recently attempted.

The SC should discard an SMS-SUBMIT or SMS-COMMAND which has the TP-RD bit set to a 1 and which has the same TP-MR value as the previous SMS-SUBMIT or SMS-COMMAND received from the same originating address. ~~The SC shall send an RP-ACK in response to~~ In the case of a ~~the~~ discarded SMS-SUBMIT or SMS-COMMAND, the SC should respond with an RP-ERROR, in which case the RP-ERROR shall include a SMS-SUBMIT-REPORT with TP-FCS indicating “SM Rejected – Duplicate SM”. In some cases, for backward compatibility with earlier phases and versions of this specification, the SC may be configured to respond with an RP-ACK.

The SMS-STATUS-REPORT also contains a TP-Message-Reference field. The value sent to the MS shall be the same as the TP-Message-Reference value generated by the MS in the earlier SMS-SUBMIT or SMS-COMMAND to which the status report relates.

9.2.3.25 TP-Reject-Duplicates (TP-RD)

The TP-Reject-Duplicates is a 1 bit field located within bit 2 of the first octet of SMS-SUBMIT and has the following values.

Bit no. 2:

0 Instruct the SC to accept an SMS-SUBMIT for an SM still held in the SC which has the same TP-MR and the same TP-DA as a previously submitted SM from the same OA.

1 Instruct the SC to reject an SMS-SUBMIT for an SM still held in the SC which has the same TP-MR and the same TP-DA as the previously submitted SM from the same OA. In this case ~~an appropriate TP-FCS value shall be returned in the SMS-SUBMIT-REPORT~~the response returned by the SC is as specified in 9.2.3.6.

3GPP TSG-T2 #16
Sophia Antipolis, France
11-15 February 2002

T2-020302

CR-Form-v5

CHANGE REQUEST

⌘	**23.040**	**CR**	**042**	⌘ **rev**	**-**	⌘	Current version:	**5.2.0** ⌘

*For **HELP** on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network □ Core Network □

Title:	⌘	Polyphonic Extended Object		
Source:	⌘	T2		
Work item code:	⌘	MESS5-EMS	***Date:*** ⌘	Feb. 15, 2002
Category:	⌘	**B**	***Release:*** ⌘	REL-5
		Use one of the following categories: ***F*** *(correction)* ***A*** *(corresponds to a correction in an earlier release)* ***B*** *(addition of feature),* ***C*** *(functional modification of feature)* ***D*** *(editorial modification)* Detailed explanations of the above categories can be found in 3GPP TR 21.900.		*Use one of the following releases:* *2 (GSM Phase 2)* *R96 (Release 1996)* *R97 (Release 1997)* *R98 (Release 1998)* *R99 (Release 1999)* *REL-4 (Release 4)* *REL-5 (Release 5)*

Reason for change:	⌘	To define a polyphony media format for EMS
Summary of change:	⌘	The reference section has been updated to include MIDI references. The EMS specific section has been modified to add a new Extended Object data format for SP-MIDI, with details of the bits and specific restrictions in Annex E.13. Also, a new annex is added which includes guidelines for content providers.
Consequences if not approved:	⌘	An existing industry standard solution for polyphonic would not be available for EMS.

Clauses affected:	⌘	Section 2, Section 9.2.3.24.10.1.11, New Annex E.13, New Annex G		
Other specs affected:	⌘	□ Other core specifications □ Test specifications □ O&M Specifications	⌘	
Other comments:	⌘			

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://ftp.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2001-03 contains the specifications resulting from the March 2001 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

2 References

The following documents contain provisions which, through reference in this text, constitute provisions of the present document.

- References are either specific (identified by date of publication, edition number, version number, etc.) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies. In the case of a reference to a 3GPP document (including a GSM document), a non-specific reference implicitly refers to the latest version of that document *in the same Release as the present document*.

[1] 3GPP TS 01.04: "Abbreviations and acronyms".

[2] 3GPP TS 02.03: "Teleservices supported by a GSM Public Land Mobile Network (PLMN)".

[3] 3GPP TS 22.004: "General on supplementary services".

[4] 3GPP TS 22.041: "Operator Determined Barring (ODB)".

[5] 3GPP TS 23.002: "Network architecture".

[6] 3GPP TS 23.008: "Organization of subscriber data".

[7] 3GPP TS 23.011: "Technical realization of supplementary services".

[8] 3GPP TS 23.015: "Technical realization of Operator Determined Barring (ODB)".

[9] 3GPP TS 23.038: "Alphabets and language-specific information".

[10] 3GPP TS 23.041: "Technical realization of Cell Broadcast Service (CBS)".

[11] 3GPP TS 43.047: "Example protocol stacks for interconnecting Service Centre(s) (SC) and Mobile-services Switching Centre(s) (MSC)".

[12] 3GPP TS 44.008: "Mobile radio interface layer 3 specification".

[13] 3GPP TS 24.011: "Point-to-Point (PP) Short Message Service (SMS) support on mobile radio interface".

[14] 3GPP TS 27.005: "Use of Data Terminal Equipment - Data Circuit terminating Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS)".

[15] 3GPP TS 29.002: "Mobile Application Part (MAP) specification".

[16] 3GPP TS 51.011: "Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface".

[17] CCITT Recommendation E.164 (Blue Book): "The international public telecommunication numbering plan".

[18] CCITT Recommendation E.163 (Blue Book): "Numbering plan for the international telephone service".

[19] CCITT Recommendation Q.771: "Specifications of Signalling System No.7; Functional description of transaction capabilities".

[20] CCITT Recommendation T.100 (Blue Book): "International information exchange for interactive videotex".

[21] CCITT Recommendation T.101 (Blue Book): "International interworking for videotex services".

[22] CCITT Recommendation X.121 (Blue Book): "International numbering plan for public data networks".

[23] CCITT Recommendation X.400 (Blue Book): "Message handling services: Message handling system and service overview".

[24] ISO/IEC10646: "Universal Multiple-Octet Coded Character Set (USC); UCS2, 16 bit coding".

[25] 3GPP TS 22.022: "Personalisation of Mobile Equipment (ME); Mobile functionality specification".

[26] 3GPP TS 23.042: "Compression Algorithm for Text Messaging Services".

[27] 3GPP TS 23.060: "General Packet Radio Service (GPRS); Service description; Stage 2".

[28] 3GPP TS 43.048: "Security Mechanisms for the SIM application toolkit; Stage 2".

[29] 3GPP TR 21.905: "Vocabulary for 3GPP Specifications".

[30] 3GPP TS 31.102: "Characteristics of the USIM application".

[31] 3GPP TS 31.101: "UICC – Terminal interface; Physical and logical characteristics".

[32] 3GPP TS 22.105: "Services and Service Capabilites".

[33] Infrared Data Association. Specifications for Ir Mobile Communications (IrMC).
iMelody.

[34] IETF RFC 822: "Standard for the format of ARPA Internet text messages".

[35] void

[36] "vCard - The Electronic Business Card", version 2.1,The Internet Mail Consortium (IMC), September 18, 1996,
URL:http://www.imc.org/pdi/vcard-21.doc".

[37] "vCalendar - the Electronic Calendaring and Scheduling Format", version 1.0,
The Internet Mail Consortium (IMC), September 18, 1996,
URL:http://www.imc.org/pdi/vcal-10.doc

[38] Scalable Polyphony MIDI Specification, MIDI Manufacturers Association (2002); http://www.midi.org

[39] Scalable Polyphony MIDI Device 5-to-24 Note Profile for 3GPP, MIDI Manufacturers Association (2002); http://www.midi.org

[40] The Complete MIDI 1.0 Detailed Specification, Incorporating all Recommended Practices, MIDI Manufacturers Association, Document version 96.1, 1996; http://www.midi.org

• • • • • •

• • • • • •

9.2.3.24.10.1.11 Extended Object

The Extended Object allows an extended code range for format types. The Extended Object may extend across segment boundaries of a concatenated short message. Octets 1 through 7 of the first Extended Object IE shall be contained in a single segment. A single segment may include one or more Extended Object IEs.

If multiple SMs are concatenated and at least one of them contains an Extended Object information element, then concatenation of the SMs shall be done using the 'Concatenated short messages, 16-bit reference number', verses the

'Concatenated short messages, 8-bit reference number' information element. The re-assembly of the Extended Object segments shall be done according to the sequence number of the associated Concatenation IE.

One or more Extended Objects may be compressed using a compression algorithm as indicated in the Compression Control IE (see clause 9.2.3.24.10.1.13).

An SME implementing the Extended Object IE shall be capable of interpreting an uncompressed concatenated message composed of at least **min_eo_msg** short messages which have been received. According to current content provider requirements and handset manufacturer constraints, variable **min_eo_msg** is set to 8.

The first Extended Object IE of an Extended Object contains a reference number, length, control data, type and position. The subsequent Extended Object IEs shall only contain Extended Object data as illustrated in Figure 9.2.24.10.11.

The IE length is variable.

Octet 1 Extended Object reference number.
A modulo 256 counter indicating the reference number for the Extended Object. Two different Extended Objects in a single concatenated message shall have different reference numbers.

Octet 2..3 Extended Object length in number of octets (integer representation) as shown in Figure 9.2.3.24.10.1.11.

Octet 4 Control data.

Bit 0 Object distribution

0 Object may be forwarded
1 Object shall not be forwarded by SMS

Bit 1 User Prompt Indicator

0 Object shall be handled normally
1 Object shall be handled as a User Prompt (see 9.2.3.24.10.1.10)

Bit 2..7 reserved

Any reserved values shall be set to 0.

Octet 5 Extended Object Type.
This octet indicates the format of the Extended Object from the table below.
If the value is reserved or if the associated format is not supported then the receiving entity shall ignore the Extend Object.

Format Type	Format Description
0x00	Predefined sound as defined in annex E.
0x01	iMelody as defined in annex E.
0x02	Black and white bitmap as defined in annex E.
0x03	2-bit greyscale bitmap as defined in annex E.
0x04	6-bit colour bitmap as defined in annex E.
0x05	Predefined animation as defined in annex E.
0x06	Black and white bitmap animation as defined in annex E.
0x07	2-bit greyscale bitmap animation as defined in annex E.
0x08	6-bit colour bitmap animation as defined in annex E.
0x09	vCard as defined in annex E.
0x0A	vCalendar as defined in annex E.
0x0B	Polyphonic melody as defined in annex E.
0x0C~~B~~.. 0xFE	Reserved
0xFF	Data Format Delivery Request as defined in annex E.

Octet 6..7 Extended Object Position (integer representation).
The Extended Object Position indicates the absolute character position within the message text after which the object shall be played or displayed. The absolute character position relates to the entire text within the concatenated message, the first character is numbered character 1.

NOTE: Although this is an absolute value, for concatenated messages, it is suggested the positions used are those that lie within the text of short message segments that have the sequence number equal to or higher than the one that contains the Extended Object IE.

If more than one Extended Object is located at the same position then they may be played or displayed in sequence or simultaneously.

Octet 8..n Extended Object Data.
This sequence of octets is structured as illustrated in the figure below and defined annex E. This figure illustrates the construction of a number of SMs containing a large Extended Object which crosses a SM boundary and is encoded into 2 SM TPDUs. The figure illustrates only the User Data field of the SM (TPDUs). For a description of concatenation of SM refer to Figures 9.2.3.24 (a, b and c)

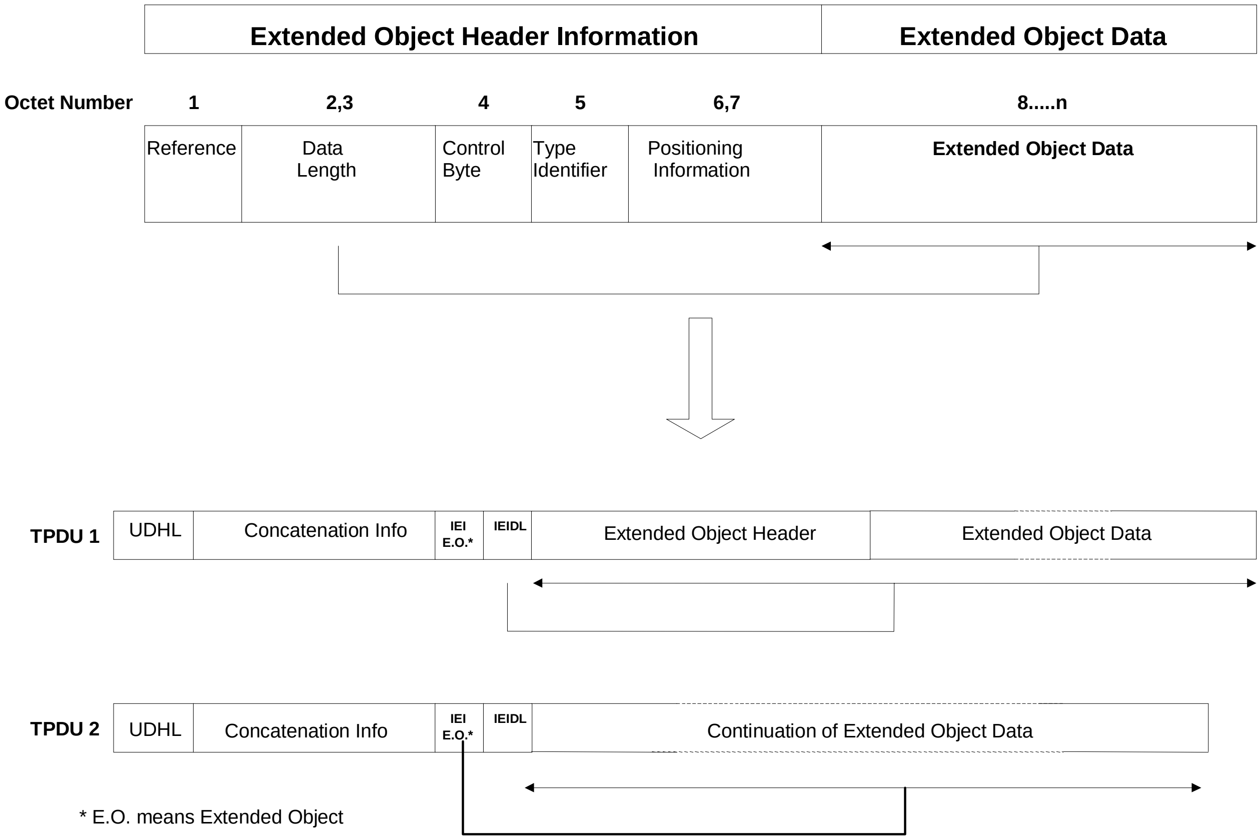


Figure 9.2.3.24.10.1.11

E.12 Data Format Delivery Request

This Data Format Delivery Request is an optional feature used by an SME to indicate which Extended Object data formats, listed in clause 9.2.3.24.10.1.11, it is requesting for delivery. This Data Format Delivery Request may be included by an SME in a MO SM containing other EMS related data, or in a MO SM independently. Processing of this data format is optional in a MT short message.

The information in this data format represents an extensible bit field with the first bit being mapped to the first Extended Object (EO) data format defined in the table in clause 9.2.3.24.10.1.11.

Octet 8

Bit 0: If set to 1 indicates support for EO data format 00

Bit 1: If set to 1 indicates support for EO data format 01

Bit 2: If set to 1 indicates support for EO data format 02

……

……

Octet n

Bit 0: If set indicates support for EO data format ((n – 8) * 8)

Bit 1: If set indicates support for EO data format ((n – 8) * 8) + 1

Bit 2: If set indicates support for EO data format ((n – 8) * 8) + 2

…….

Any unused bits in the last octet shall be set to zero.

E.13 Polyphonic melody

A Polyphonic melody can be integrated as an extended object in one or more short messages. Informative guidelines for the creation of polyphony content using SP-MIDI [38] are listed in Annex G.

However, in order to guarantee the interoperability with legacy mobile devices which are not able to interpret specific SP-MIDI content, the following considerations shall be taken into account for content creation:

- When content is not provided in SP-MIDI format the presence of the MIP table in polyphonic extended objects is not mandatory. Since a receiving SME supporting polyphonic extended objects may decide to ignore and skip the content of a MIP message by implementing its own note stealing or channel masking strategy when played. However, when SP-MIDI format data is present and the message is stored and subject to potential forwarding, the specific SP-MIDI content shall be kept as received by the SME.
- the additional rhythm channel as specified in section 3.2 in [38] might not be supported by the receiving SME.

Note: SP-MIDI is not available yet as a published specification by MMA/AMEI. In case this specification could not be available before the date the present release is frozen, SMF type 0 or 1 is proposed as the polyphonic format to be used for the creation of polyphonic content in EMS in accordance with [40].

Octet 8..n SMF as defined in [38], [40]

…..

…..

Annex G (informative): Development Guidelines for Creation of Polyhony Using SP-MIDI

While Scalable Polyphony-MIDI (SP-MIDI) [38] is a full-featured standard for synthesizing music, using a few guidelines SP-MIDI [38, 39] can be optimized for wireless devices. These guidelines can be grouped as optimizing individual notes, and to minimize the overall size of a melody.

Note: SP-MIDI profile for 3GPP [39] is not available yet as a published specification by MMA/AMEI and reference to 3GPP not validated by 3GPP MCC in accordance with its liaison rules. In case this specification could not be available before the date the present release is frozen, the guidelines to be used by content providers are those following that are independant of the specifiation mentioned in this present note.

1. Running status

In the Musical Instrument Digital Interface (MIDI) standard, a key-on or a key-off event will use at most three bytes each, cf. [40], However, in case several key events occur on the same MIDI-channel, running status can be used. In principle running status means that the first byte of, e.g. key-on is omitted. In addition, the key-on event having a velocity of zero is equivalent to the key-off event. Thus, combining running status and using key-on with zero velocity, as the key-off event will reduce the number of bytes needed to encode key events.

EXAMPLE:

Without running status the sequence

91 2E 23 8E, 91 2B 50 8E, 81 2E 64 00, 81 2B 64 00

means "Key 2E ON" Velocity 23 MIDI Ch 1", "Key 2B ON Velocity 50 MIDI Ch 1", "Key 2E OFF Velocity 64 MIDI Ch 1", "Key 2B OFF Velocity 64 MIDI Ch 1". Using running status will reduce the sequence into

91 2E 23 8E, 2B 50 8E, 2E 00 00, 2B 00 00,

That is, the command byte is omitted and velocity zero is used for key off.

2. File type considerations

The SP-MIDI content can be stored in, a Standard MIDI File (SMF) of type 0 or type 1 [40]. In a type 0 SMF, one header chunk and one track chunk is used. In a type 1 SMF one header chunk and several track chunks are used. SMF type 2 should not be used

3. File size reduction

In general it is more efficient to store the MIDI data as a type 1 file. The increased efficiency is reached if each track contains one MIDI channel and one instrument (This is often the case). Evidently, running status can be applied on each individual track reducing the track size. To further reduce the size of the file use one track per used MIDI channel. That is, if a temple/conductor track exists merge it with the first instrument track. Remove, all meta events which are not necessary, e.g. "track name", "lyric". To summarize, the following measures can be taken in order to reduce the SMF:

1. Use SMF type 1 (Or check if type 1 is smaller than type 0 and use the smallest)
2. Use running status
3. One and only one instrument per track. Try not to change channels.
4. Do not change tempo in the middle of the music, i.e., only set tempo once.
5. Use beat, instead of SMPTE, to set tempo
6. Copyright is on automatically
7. Remove controller messages, which are optional according to [39].
8. Turn off the options below:
    - Sequence Number - MIDI sequence ids
    - Text - embedded text for anything
    - Sequence / Track Name
    - Instrument Name
    - Lyric
    - Marker - for synchronization purposes
    - Cue Point
    - Midi Channel Presix - associate channels with all events following

- Sequencer-Specific settings

Items 1 to 3 above optimize the notes, while items 4 to 8 optimize the overall melody. The above measures will provide an SMF, which is ready for compression. However, prior to compression the composer/content author can consider to use few values for key velocity and thereby increasing the redundancy of the file.

4. Restrictions

Content creators should not expect the full support for the following features:

- MIDI message channel pressure
- MIDI message pitch bend
- Individual stereophonic panoramic (pan) as expressed in table 5 in [39].
- MIDI message master volume

Content creators should not expect a time granularity better than 5ms to be supported by the SME.

To ensure interoperability, the first value of the MIP table should be no more than 6 voices.

3GPP TSG-T2 #16 **_T2-020301_**

Sophia Antipolis, France, February 11-15, 2001

CR-Form-v5

CHANGE REQUEST

⌘ **23.040 CR 041** ⌘ rev ⌘ Current version: **5.2.0** ⌘

For ***HELP*** *on using this form, see bottom of this page or look at the pop-up text over the ⌘ symbols.*

Proposed change affects: ⌘ (U)SIM □ ME/UE ☒ Radio Access Network □ Core Network □

Title:	⌘	Wireless Vector Graphics in EMS		
Source:	⌘	T2		
Work item code:	⌘	MESS5-EMS	***Date:*** ⌘	February 2002
Category:	⌘	**B**	***Release:*** ⌘	REL-5

Use one of the following categories:
F *(essential correction)*
A *(corresponds to a correction in an earlier release)*
B *(Addition of feature),*
C *(Functional modification of feature)*
D *(Editorial modification)*
Detailed explanations of the above categories can
be found in 3GPP TR 21.900.

Use one of the following releases:

2	*(GSM Phase 2)*
R96	*(Release 1996)*
R97	*(Release 1997)*
R98	*(Release 1998)*
R99	*(Release 1999)*
REL-4	*(Release 4)*
REL-5	*(Release 5)*

Reason for change: ⌘ WVG (Wireless Vector Graphics) provides a scalable and animated picture capability to EMS with high efficiency for message size by utilising a combination of compression technologies.

WVG also provides Character Size drawing (or glyph) capability to EMS, which allows complex Asian language messages to be handwritten and displayed efficiently.

Summary of change: ⌘ Two new IE types and one new extended object type added.

Consequences if not approved: ⌘ Inefficient large bitmap picture messages, even with certain levels of compression implemented in EMS, may cause delays on handsets and overload on the network. . In addition, bitmap images do not scale well in size and suffer from “jaggies” when resized for different screen resolutions.

Without an efficient animated vector graphics message type, EMS may lose its time window in the marketplace.

Clauses affected: ⌘ 3.10.6, 9.2.3.24, 9.2.3.24.10.1.11, 9.2.3.24.10.1.12, 9.2.3.24.10.1.11, Annex E.13, Annex G (should also change existing Annex G to Annex H)

Other specs affected: ⌘
□ Other core specifications ⌘
□ Test specifications
□ O&M Specifications

Other comments: ⌘ None

How to create CRs using this form:

Comprehensive information and tips about how to create CRs can be found at: http://www.3gpp.org/3G_Specs/CRs.htm. Below is a brief summary:

1) Fill out the above form. The symbols above marked ⌘ contain pop-up help information about the field that they are closest to.

2) Obtain the latest version for the release of the specification to which the change is proposed. Use the MS Word "revision marks" feature (also known as "track changes") when making the changes. All 3GPP specifications can be downloaded from the 3GPP server under ftp://www.3gpp.org/specs/ For the latest version, look for the directory name with the latest date e.g. 2000-09 contains the specifications resulting from the September 2000 TSG meetings.

3) With "track changes" disabled, paste the entire CR form (use CTRL-A to select it) into the specification just in front of the clause containing the first piece of changed text. Delete those parts of the specification which are not relevant to the change request.

3.10.5 vCard and vCalendar

A message may contain vCard and vCalendar objects as specified in [36][37]. These may be transmitted in a compressed form.

3.10.6 WVG (Wireless Vector Graphics) Object

A message may contain one or more WVG objects. A WVG object is a vector graphics picture or animation and is scalable. Two subtypes of WVG objects are supported; Standard WVG object and Character Size WVG object. Actual display size of a Standard WVG object depends on display screen size and MMI implementation on terminals. A Character Size WVG object has a height that equals or is similar to the height of message text but with variable width. Character Size WVG object may be edited in the same way as standard text, e.g. insertion deletion and text wrapping.

3.10.6.1 Overview of WVG Graphical Primitives

The WVG element is used to describe vector graphics objects. The vector graphics format is used to allow the creation of small pictures which may include simple animation or the creation small handwritten sketches. WVG makes use of the following graphical primitives (full detail is listed in annex G.2) These primitives can be used to describe a compact drawing.

List of Graphical Primitives

- Polylines (G2.1)
  - Simple Line Polyline (G.2.1.1)
  - Circular Polyline (G.2.1.2)
  - Bezier lines (G.2.1.3)
- Polygons (G.2.2)
  - Arbitrary Polygon (G.2.2)
  - Regular Polygon (G.2.4)
  - Star Shaped Polygon (G.2.4)
  - Regular Grid Element (G.2.4)
- Ellipses (G.2.3.1)
- Rectangles (G.2.3.2)
- Text Element (G.2.5)
- Grouping Element (G.2.6)
- Reuse Element (G.2.7)
- Animations Elements (G.2.8)
- Frame Element (G.2.9)
- Local Element (G.2.10)

1. 9.2.3.24 TP-User Data (TP-UD)

The length of the TP-User-Data field is defined in the PDU's of the SM-TL (see clause 9.2.2).
The TP-User-Data field may comprise just the short message itself or a Header in addition to the short message depending upon the setting of TP-UDHI.
Where the TP-UDHI value is set to 0 the TP-User-Data field comprises the short message only, where the user data can be 7 bit (default alphabet) data, 8 bit data, or 16 bit (UCS2 [24]) data.
Where the TP-UDHI value is set to 1 the first octets of the TP-User-Data field contains a Header in the following order starting at the first octet of the TP-User-Data field.
Irrespective of whether any part of the User Data Header is ignored or discarded, the MS shall always store the entire TPDU exactly as received.

FIELD	LENGTH
Length of User Data Header	1 octet
Information-Element-Identifier "A"	1 octet
Length of Information-Element "A"	1 octet
Information-Element "A" Data	0 to "n" octets
Information-Element-Identifier "B"	1 octet
Length of Information-Element "B"	1 octet
Information-Element "B" Data	0 to "n" octets
Information-Element-Identifier "X"	1 octet
Length of Information-Element "X"	1 octet
Information-Element "X" Data	0 to "n" octets

The diagram below shows the layout of the TP-User-Data-Length and the TP-User-Data for uncompressed GSM 7 bit default alphabet data. The UDHL field is the first octet of the TP-User-Data content of the Short Message.

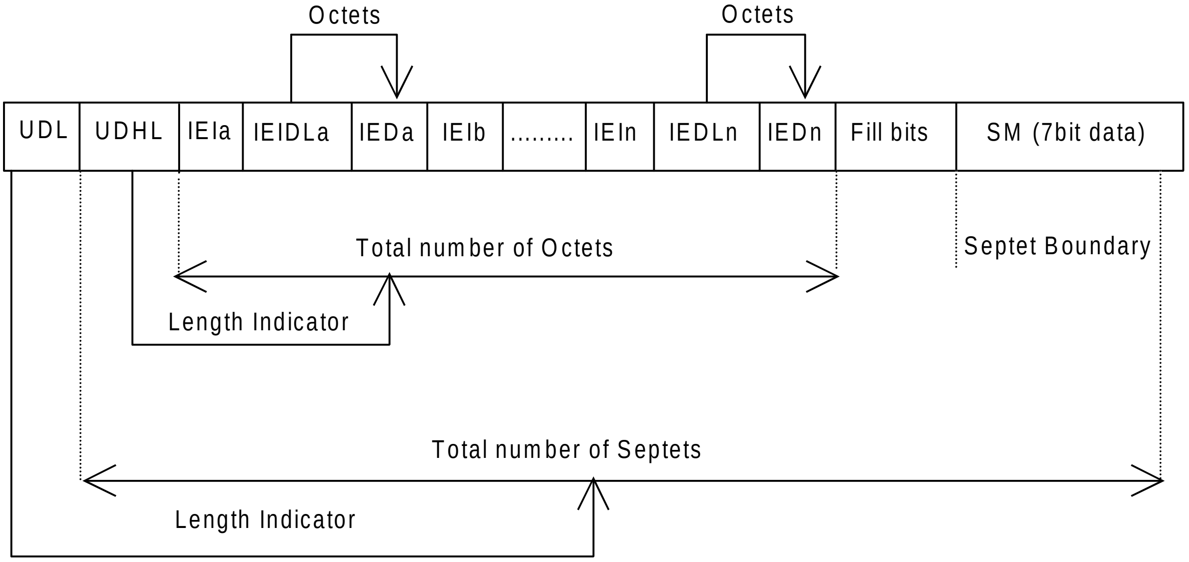


Figure 9.2.3.24 (a)

The diagram below shows the layout of the TP-User-Data-Length and the TP-User-Data for uncompressed 8 bit data or uncompressed UCS2 data. The UDHL field is the first octet of the TP-User-Data content of the Short Message.

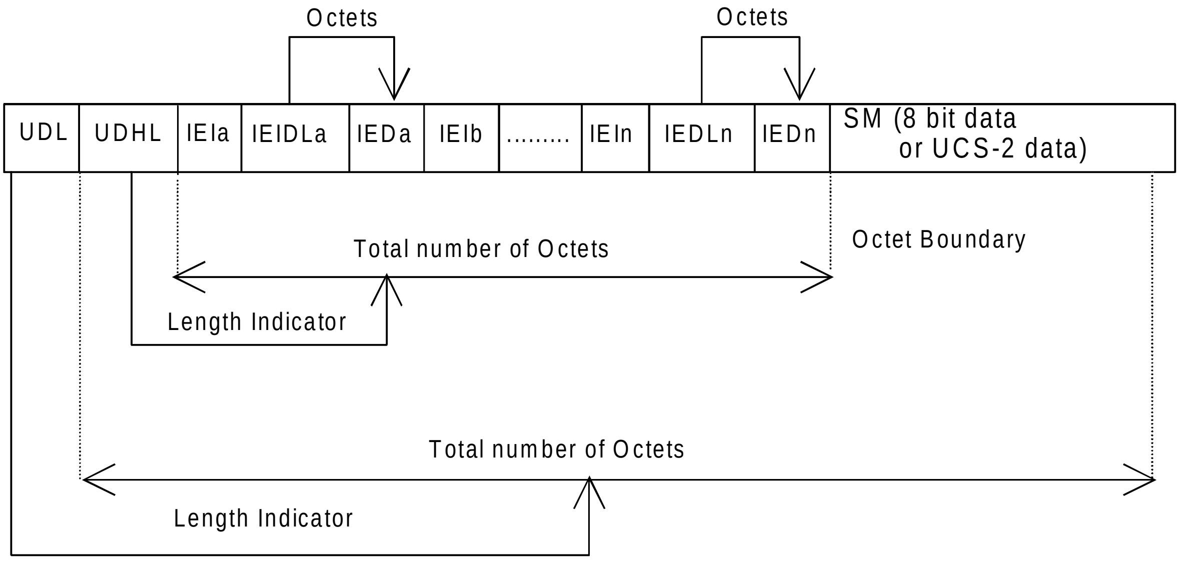


Figure 9.2.3.24 (b)

The diagram below shows the layout of the TP-User-Data-Length and the TP-User-Data for compressed GSM 7 bit default alphabet data, compressed 8 bit data or compressed UCS2 data. The UDHL field is the first octet of the TP-User-Data content of the Short Message.

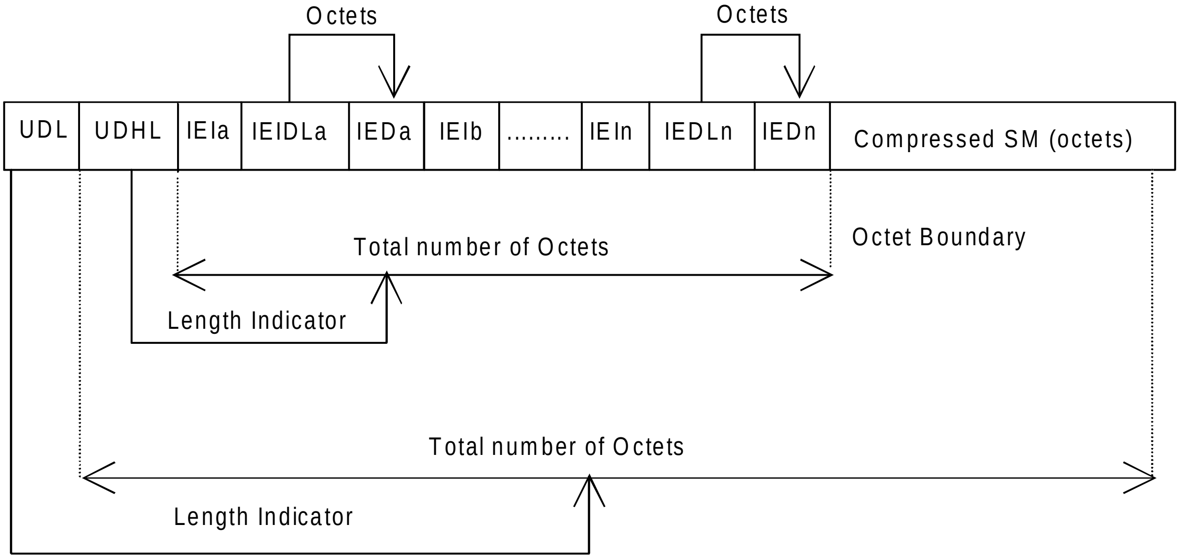


Figure 9.2.3.24 (c)

The definition of the TP-User-Data-Length field which immediately precedes the "Length of User Data Header" is unchanged and shall therefore be the total length of the TP-User-Data field including the Header, if present. (see 9.2.3.16).
The "Length-of-Information-Element" fields shall be the integer representation of the number of octets within its associated "Information-Element-Data" field which follows and shall not include itself in its count value.
The "Length-of-User-Data-Header" field shall be the integer representation of the number of octets within the "User-Data-Header" information fields which follow and shall not include itself in its count or any fill bits which may be present (see text below).
Information Elements may appear in any order and need not follow the order used in the present document. Information Elements are classified into 3 categories as described below.

- SMS Control – identifies those IEIs which have the capability of dictating SMS functionality.
- EMS Control – identifies those IEIs which manage EMS Content IEIs.
- EMS Content – identifies those IEIs containing data of a unique media format.

It is permissible for certain IEs to be repeated within a short message, or within a concatenated message. There is no restriction on the repeatability of IEs in the EMS Content classification. The repeatability of SMS Control and EMS Control IEs is determined on an individual basis. See the IE table below for the repeatability of each IE.

In the event that IEs determined as not repeatable are duplicated, the last occurrence of the IE shall be used. In the event that two or more IEs occur which have mutually exclusive meanings (e.g. an 8bit port address and a 16bit port address), then the last occurring IE shall be used.

If the length of the User Data Header is such that there are too few or too many octets in the final Information Element then the whole User Data Header shall be ignored.

If any reserved values are received within the content of any Information Element then that part of the Information Element shall be ignored.

The Information Element Identifier octet shall be coded as follows:

VALUE (hex)	MEANING	Classification	Repeatability
00	Concatenated short messages, 8-bit reference number	SMS Control	No
01	Special SMS Message Indication	SMS Control	Yes
02	Reserved	N/A	N/A
03	Value not used to avoid misinterpretation as <LF> character	N/A	N/A
04	Application port addressing scheme, 8 bit address	SMS Control	No
05	Application port addressing scheme, 16 bit address	SMS Control	No
06	SMSC Control Parameters	SMS Control	No
07	UDH Source Indicator	SMS Control	Yes
08	Concatenated short message, 16-bit reference number	SMS Control	No
09	Wireless Control Message Protocol	SMS Control	Note 3
0A	Text Formatting	EMS Control	Yes
0B	Predefined Sound	EMS Content	Yes
0C	User Defined Sound (iMelody max 128 bytes)	EMS Content	Yes
0D	Predefined Animation	EMS Content	Yes
0E	Large Animation (16*16 times 4 = 32*4 =128 bytes)	EMS Content	Yes
0F	Small Animation (8*8 times 4 = 8*4 =32 bytes)	EMS Content	Yes
10	Large Picture (32*32 = 128 bytes)	EMS Content	Yes
11	Small Picture (16*16 = 32 bytes)	EMS Content	Yes
12	Variable Picture	EMS Content	Yes
13	User prompt indicator	EMS Control	Yes
14	Extended Object	EMS Content	Yes
15	Reused Extended Object	EMS Control	Yes
16	Compression Control	EMS Control	No
17	Object Distribution Indicator	EMS Control	Yes
18	Standard WVG object	EMS Content	Yes
19	Character Size WVG object	EMS Content	Yes
~~18~~1A-1F	Reserved for future EMS features (see subclause 3.10)	N/A	N/A
20	RFC 822 E-Mail Header	SMS Control	No
21	Hyperlink format element	SMS Control	Yes
22 – 6F	Reserved for future use	N/A	N/A
70 – 7F	(U)SIM Toolkit Security Headers	SMS Control	Note 1
80 – 9F	SME to SME specific use	SMS Control	Note 2
A0 – BF	Reserved for future use	N/A	N/A
C0 – DF	SC specific use	SMS Control	Note 2
E0 – FF	Reserved for future use	N/A	N/A

Note 1: The functionality of these IEIs is defined in 3GPP TSG 23.048 [28], and therefore, the repeatability is not within the scope of this document and will not be determined here.

Note 2: The functionality of these IEIs is used in a proprietary fashion by different SMSC vendors, and therefore, are not within the scope of this technical specification.

Note 3: The functionality of these IEIs is defined by the WAP Forum and therefore the repeatability is not within the scope of this document and will not be determined here.

A receiving entity shall ignore (i.e. skip over and commence processing at the next information element) any information element where the IEI is Reserved or not supported. The receiving entity calculates the start of the next information element by looking at the length of the current information element and skipping that number of octets.

9.2.3.24.10.1.10 User Prompt Indicator

With the User Prompt Indicator a sending entity is able to indicate to the receiving entity, that the following object is intended to be handled at the time of reception, e.g. by means of user interaction. The object may be a picture, an animation, a User Defined Sound or a combination of these.

For example the User Prompt Indicator may be used when sending an operators logo to the ME that should be displayed instead of the operators name in standby mode.

When receiving the object the user shall be prompted to accept or discard the object. After this user interaction the SM may be discarded.

The User Prompt Indicator IE shall immediately precede the corresponding object IE(s).

If a User Prompt Indicator IE is not followed by a corresponding object IE it shall be discarded.

The Information-Element-Data octet(s) shall be coded as follows.

Octet 1 Number of corresponding objects

This octet shall contain the number of corresponding objects as an integer value.

Where Octet 1 indicates that the User Prompt Indicator refers to more than one object, the ME should check the validity of the objects referenced for stitching together. The objects should be considered for stitching if they are either Images (Small, Large, Variable Pictures) or User Defined Sounds, and all of the objects referenced by the User Prompt Indicator IE are of the same type. Animations, Text formatting and pre-defined sound IE's are not suitable for stitching.

User defined sounds may be stitched by concatenating the data contained within each User Defined Sound IE into a single melody object, this may be achieved by ignoring the iMelody header and footer information of the second and subsequent User Defined Sound IE's referenced from the User Prompt Indicator.

Images may be joined along their vertical edges, to form a single "wide" image, the resulting image will have a width equal to the sum of the widths of all the images defined in the User Prompt Indicator.

9.2.3.24.10.1.11 Standard WVG Object

The Standard WVG object as defined by IEI 18 is structured as follows:

Octet 1 position indicating in the SM data the instant the object shall be displayed in the SM data

Octet 2..n Standard WVG object bit stream

The unused bits in the last octet will be filled with 0

The detailed data format and attributes of Standard WVG object are defined in Annex G.

The bit order is defined as follows:

The octet with a smaller octet number stores the bits appearing in the front position in the bit stream; the most significant bit in an octet stores the first bit in position in a 8-bit segment in the bit stream.

A Standard WVG object may or may not have fixed size. In either case, display size should be determined by the terminal implementation. Recommended display size is a largest possible size on terminal screen while aspect ratio shall be maintained.

9.2.3.24.10.1.12 Character Size WVG Object

The Character Size WVG object as defined by IEI 19 is structured as follows:

Octet 1 position indicating in the SM data the instant the object shall be displayed in the SM data

Octet 2..n Character Size WVG bit stream

The unused bits in the last octet will be filled with 0

The detailed data format and attributes of Character Size WVG object are defined in Annex G.

The bit order is defined as follows:

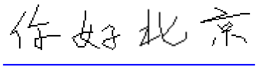
The octet with a smaller octet number stores the bits appearing in the front position in the bit stream; the most significant bit in an octet stores the first bit in position in a 8-bit segment in the bit stream.

A Character Size WVG object is a small graphics similar to the size of a typed character. The display height for a Character Size WVG object is decided by the terminal implementation. Recommended Character Size WVG object height is to be similar to the message text font height. The width of a Character Size WVG object is variable depending on the aspect ratio defined in the object. Character Size WVG objects can appear more than one time in one message.

Example:

Dad, I ♡ you!

In the above example, the "heart" is a Character Size WVG object at the position in between the letter "I" and "y".



In the above example, there are 4 Character Size WVG objects, each representing a Chinese character.

9.2.3.24.10.1.~~11~~13 Extended Object

The Extended Object allows an extended code range for format types. The Extended Object may extend across segment boundaries of a concatenated short message. Octets 1 through 7 of the first Extended Object IE shall be contained in a single segment. A single segment may include one or more Extended Object IEs.
If multiple SMs are concatenated and at least one of them contains an Extended Object information element, then concatenation of the SMs shall be done using the 'Concatenated short messages, 16-bit reference number', verses the 'Concatenated short messages, 8-bit reference number' information element. The re-assembly of the Extended Object segments shall be done according to the sequence number of the associated Concatenation IE.
One or more Extended Objects may be compressed using a compression algorithm as indicated in the Compression Control IE (see clause 9.2.3.24.10.1.13).
An SME implementing the Extended Object IE shall be capable of interpreting an uncompressed concatenated message composed of at least **min_eo_msg** short messages which have been received. According to current content provider requirements and handset manufacturer constraints, variable **min_eo_msg** is set to 8.

The first Extended Object IE of an Extended Object contains a reference number, length, control data, type and position. The subsequent Extended Object IEs shall only contain Extended Object data as illustrated in Figure 9.2.24.10.11.
The IE length is variable.

Octet 1 Extended Object reference number.
A modulo 256 counter indicating the reference number for the Extended Object. Two different Extended Objects in a single concatenated message shall have different reference numbers.

Octet 2..3 Extended Object length in number of octets (integer representation) as shown in Figure 9.2.3.24.10.1.11.

Octet 4 Control data.

Bit 0 Object distribution

0 Object may be forwarded
1 Object shall not be forwarded by SMS
Bit 1 User Prompt Indicator
0 Object shall be handled normally
1 Object shall be handled as a User Prompt (see 9.2.3.24.10.1.10)
Bit 2..7 reserved
Any reserved values shall be set to 0.

Octet 5 Extended Object Type.
This octet indicates the format of the Extended Object from the table below.
If the value is reserved or if the associated format is not supported then the receiving entity shall ignore the Extend Object.

Format Type	Format Description
0x00	Predefined sound as defined in annex E.
0x01	iMelody as defined in annex E.
0x02	Black and white bitmap as defined in annex E.
0x03	2-bit greyscale bitmap as defined in annex E.
0x04	6-bit colour bitmap as defined in annex E.
0x05	Predefined animation as defined in annex E.
0x06	Black and white bitmap animation as defined in annex E.
0x07	2-bit greyscale bitmap animation as defined in annex E.
0x08	6-bit colour bitmap animation as defined in annex E.
0x09	vCard as defined in annex E.
0x0A	vCalendar as defined in annex E.
0x0B	Standard WVG object as defined in annex E
~~0x0B~~0x0C.. 0xFE	Reserved
0xFF	Data Format Delivery Request as defined in annex E.

Octet 6..7 Extended Object Position (integer representation).
The Extended Object Position indicates the absolute character position within the message text after which the object shall be played or displayed. The absolute character position relates to the entire text within the concatenated message, the first character is numbered character 1.

NOTE: Although this is an absolute value, for concatenated messages, it is suggested the positions used are those that lie within the text of short message segments that have the sequence number equal to or higher than the one that contains the Extended Object IE.

If more than one Extended Object is located at the same position then they may be played or displayed in sequence or simultaneously.

Octet 8..n Extended Object Data.
This sequence of octets is structured as illustrated in the figure below and defined annex E. This figure illustrates the construction of a number of SMs containing a large Extended Object which crosses a SM boundary and is encoded into 2 SM TPDUs. The figure illustrates only the User Data field of the SM (TPDUs). For a description of concatenation of SM refer to Figures 9.2.3.24 (a, b and c)

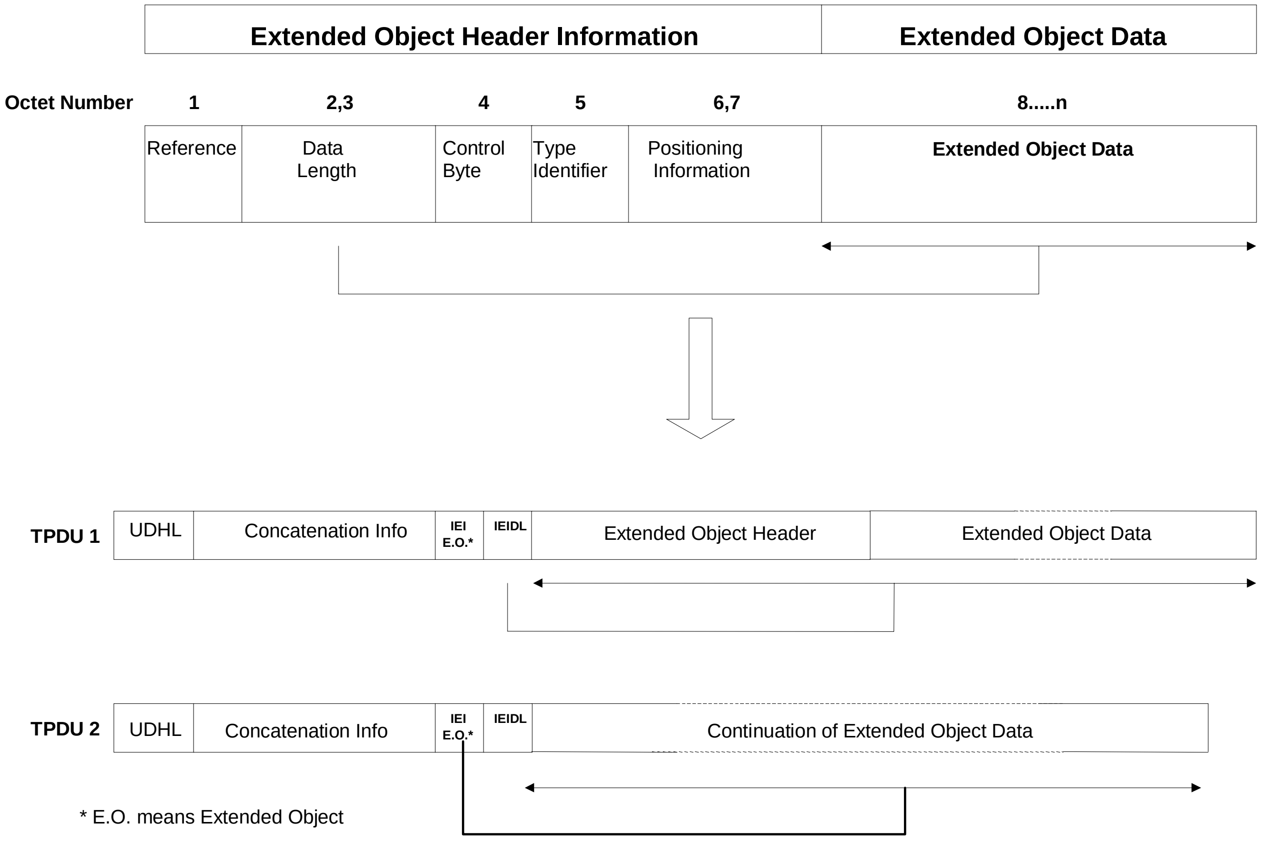


Figure 9.2.3.24.10.1.11

Annex E (normative): Extended Object Format Type

E.12 Data Format Delivery Request

This Data Format Delivery Request is an optional feature used by an SME to indicate which Extended Object data formats, listed in clause 9.2.3.24.10.1.11, it is requesting for delivery. This Data Format Delivery Request may be included by an SME in a MO SM containing other EMS related data, or in a MO SM independently. Processing of this data format is optional in a MT short message.

The information in this data format represents an extensible bit field with the first bit being mapped to the first Extended Object (EO) data format defined in the table in clause 9.2.3.24.10.1.11.

Octet 8

Bit 0: If set to 1 indicates support for EO data format 00
Bit 1: If set to 1 indicates support for EO data format 01
Bit 2: If set to 1 indicates support for EO data format 02
……
……

Octet n

Bit 0: If set indicates support for EO data format ((n – 8) * 8)
Bit 1: If set indicates support for EO data format ((n – 8) * 8) + 1
Bit 2: If set indicates support for EO data format ((n – 8) * 8) + 2
…….

Any unused bits in the last octet shall be set to zero.

E.13 Standard WVG Object

The Standard WVG object as defined by Format Type 0x0B in the Extended Object IE is as follows.

Octet 8..n Standard WVG object bit stream

The unused bits in the last octet will be filled with 0

The detailed data format and attributes of Standard WVG object are defined in Annex G.

The bit order is defined as follows:

The octet with a smaller octet number stores the bits appearing in the front position in the bit stream; the most significant bit in an octet stores the first bit in position in a 8-bit segment in the bit stream.

A Standard WVG object may or may not have fixed size. In either case, display size should be determined by the terminal implementation. Recommended display size is a largest possible size on terminal screen while aspect ratio shall be maintained.

Annex G (Normative): WVG (Wireless Vector Graphics) Data Format

WVG (Wireless Vector Graphics) is a compact binary data format for vector graphics. WVG data is represented by a bit stream, composed of a header, codec parameters and graphical elements. The bit representation of the drawing and contained graphical elements is designed such that the bit stream can be optimized for smallest possible size.

G.1 Introduction

G.1.1 Standard and Character Size WVG Elements

A Standard WVG element is defined by the complete WVG specification. Using a set of the WVG specification with a set of default values, a simplified vector graphics can be used to represent a simple and small vector graphics or glyph. Character Size WVG elements can be included in normal text to represent a handwritten character or symbols that are not supported by character coding system and the font library.

G.1.2 Compression Methods

A combination of compression methods is used in the WVG to achieve the best compression ratio for simple vector graphics and animations. They include:

- **Switchable Linear or Non-linear coordination system**: when graphical elements in a drawing are not evenly distributed, the representation of coordinates can be optimized using a non-linear coordinate system (uneven coordinates)
- **Bit packing**: variable number of bits to represent a number. The number of bits used in WVG can vary from 1 bit to 16 bits.
- **Local envelope**: use a dedicated coordinate system to describe elements in a small area using relatively small coordinate numbers
- **Variable resolution**: in coordinates, sizes, angles, scale and etc, different resolutions can be used for a graphical element to save the number of bits needed for representing a value.
- **Palettes**: color and element ID can be mapped using a palette defined in the drawing header. This also saves the number of bits for representing a color value and an element ID.
- **Default values**: many values can be omitted to use default values. E.g. when no color scheme is defined, the data describes a mono drawing
- **Default Animation Timing**: in addition to standard time based animation, WVG uses a simplified animation model. In Simple Animation mode, no timing is needed for describing animations. Instead, a cycle is defined to describe the timing for these animations.

G.1.3 Coordinate Systems

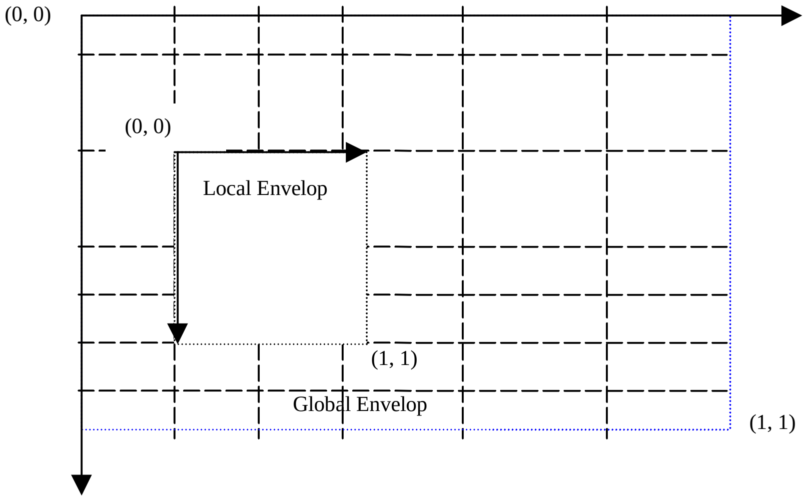
There are two coordinate systems used in WVG, namely Compact Coordinate System and Flat Coordinate System.

G.1.3.1 Compact Coordinate System

In compact coordinate system, a drawing area is defined as rectangle area called envelope. There are two types of envelopes, global envelope and local envelope. The global envelope is a base area in which the drawing is contained. There is only one global envelope. A local envelope is a square area completely or partially within the global envelop. There is no specific global envelope size specified in the data format. The physical display size is decided at rendering time.

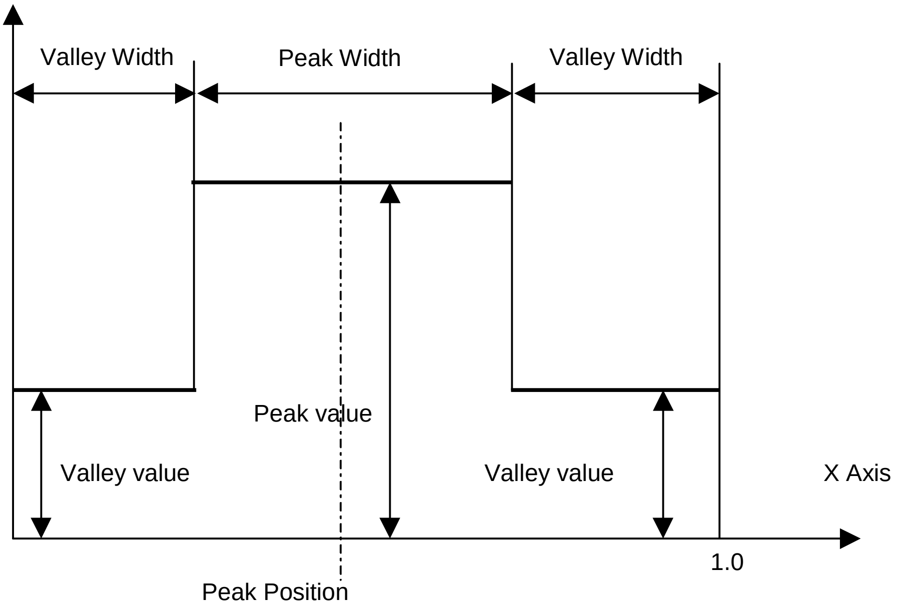
The aspect ratio and orientation are defined in the data header and should be maintained when the drawing is displayed.

Aspect ratios include 1:1, 4:3, 16:9 up to 1024: 729 (height:width), in both portrait and landscape orientation. Aspect ratio for Characters Size WVG elements only has landscape orientation.



In Compact Coordinate System, coordinates are restricted to certain positions which are the cross points of a grid. The grid is defined in the WVG data header, set by a group of parameters. The grid lines along with x axis or y axis may be unevenly distributed.

The global grid can be described using a curve shown above.



There are one peak and two valleys in the curve. The definition of the curve is:

- Peak position: the central position of a peak
- Peak value: a value equal or larger than 1.0
- Peak width: a value less than 1.0
- All valleys should have the same value
- The total area enclosed by the curve and the x-axis from 0.0 to 1.0 is always equating to 1.0

The curve can be uniquely defined by peak position, peak value and peak width. Once the parameters are determined, other values such as valley value can be calculated. Once a curve is given, grid line positions can be calculated according to the following function:

$$\int_0^{X_k} d(x)dx = \frac{k}{n-1}$$

Where X_k is the position of the k^{th} grid line, where n is total number of grid lines. d(x) is the curve function described in this document.

In Standard WVG, the curve parameters are preset as follows.

Variable parameters:

- Number of grid lines: 15, 31, 63 or 127
- Peak value: 1.0, 1.5, 2.0 and 2.5
- Peak position: 13 options from 0.0(0/12), 0.083333(1/12), 0.166667(2/12) to 1.0(12/12)
- Peak width: 0.3, 0.4, 0.5 and 0.6

When a portion of a peak exceeds the global envelope only the part within the global envelope is valid.

For Character Size WVG or glyph, the parameters are set as follows.

Predefined parameters:

- Peak width: 0.4

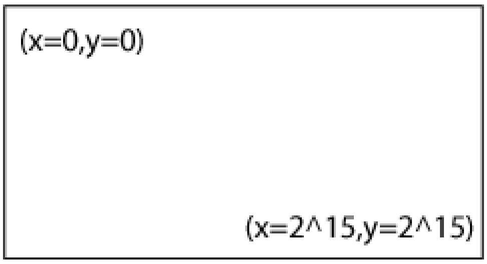
Variable parameters:

- Number of grid lines: 7, 15, 31 or 63
- Peak value: 1.0 or 1.5
- Peak position: 0.3333(1/3), 0.5, 0.6667(2/3)

When using relative coordinates in Compact Coordinate System (refer to G1.3.3), some elements may be specified with specific resolution, which is independent of the global resolution. There are 8 predefined resolutions available for "re-definition resolution", there are 1/27, 1/38, 1/48, 1/64, 1/85, 1/128 and 1/160 of the length of the shorter global envelop edge. Re-definition of resolution only applies to elements in global scope.

G.1.3.2 Flat Coordinate System

The Flat Coordinate System is a 16 bit signed coordinate system with the top left coordinate of the screen being defined as (x=0,y=0) and the bottom right coordinate being described as (x=2^15, y = 2^15). Note that this expresses the dynamic range of the coordinate system, however it does not mean that all drawings are of this size.



G.1.3.3 Coordinate Values

Coordinate values may be represented using two methods: absolute coordinate and relative coordinate.

Absolute Coordinate: an absolute coordinate is a pair of x and y coordinate number. In WVG Compact Coordinate System, absolute coordinate values are the coordinate grid line numbers and are always positive.

Relative Coordinate: the relative coordinate is used only in lines and transform. If the start point is defined by an absolute coordinate, subsequent points can be described by relative coordinates, which are relative grid units from the previous point. A relative coordinate is signed, and it may be positive or negative. A relative coordinate may be used in both global and local coordinate systems. A relative coordinate may exceed the scope of the local envelope that defines the start point of the line.

G.1.4 Color Schemes

WVG supports the following color schemes.

- Black and White (2 Colors): black and white color.
- 2-bit Grayscales: four grayscales are defined as (0,0,0), (85,85,85), (170,170,170) and (255,255,255) in 24-bit RGB color format.
- 4 Default Colors
- 6-bit RGB Color: it is similar to 24-bit RGB color definition but uses only 2 bits to represent a single color, in which value 0, 1,2 and 3 represent 8-bit color value 0, 85, 170 and 255 respectively.
- 6-bit RGB Color Using 2nd Palette
- 8-bit W3C websafe color
- 12-bit and 24-bit RGB color

There are 2 optional drawing pens in WVG, stroke pen and fill pen. Stroke pen and fill pen can be specified with one of the colors defined using the scheme. When the stroke pen is not defined, BLACK should be used for strokes. When the fill pen is not defined, no fill should be applied.

G.1.5 Rendering Model

WVG uses painter model. The elements appears in the later position in the WVG bit stream will overrides the overlapped portion of the elements which appear in the front in the bit stream.

G.2 Graphical Elements

WVG defines a set of graphical and animation elements. Among them, line, shape and text elements are the building blocks to form a drawing. These elements can be transformed, grouped and animated. There are also special elements that are auxiliary.

G.2.1 Line Elements

There are 3 types of lines: polyline, circular polyline and bezier polyline. A polyline can represent a dot when there is only a start point defined.

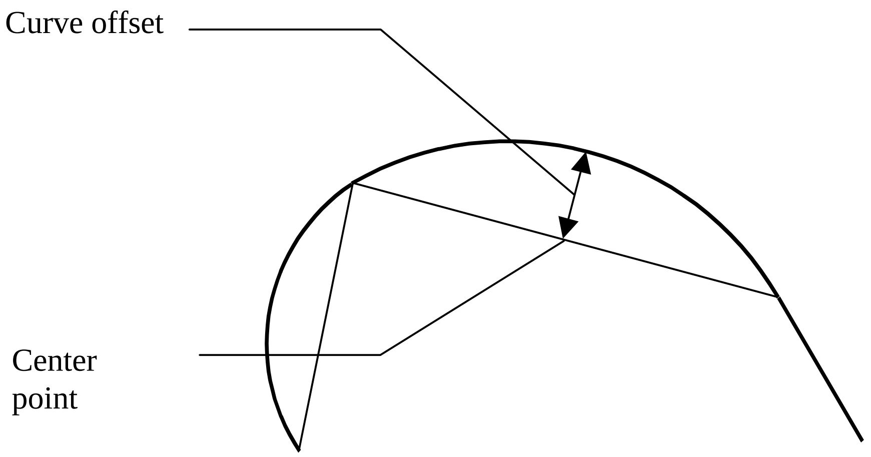
A line element has its reference point at the starting point.

G.2.1.1 Polyline

Polyline is a set of straight lines connecting a sequence of points. When there is only one point, it is defined as a Dot.

G.2.1.2 Circular Polyline

Circular Polyline is a line that contains at least one circular curve segment. The curve segment connects two adjacent points by a circular arc. The curve segment is determined by the two adjacent points and a curve offset (the perpendicular distance from the center of the line connecting the adjacent points to the circular arc).



Curve offset values are within the range – 0.5 to 0.5, inclusive. A value of 0.5 or – 0.5 identifies that the curve offset equals half of length of the connecting line. The value indicates that the curve is close to a half circle. A positive value indicates that the curve is at the left side of the base line viewed from the curve direction. A negative value indicates that the curve is at the right side of the base line viewed from the curve direction.

G.2.1.3 Bezier Polyline

A Bezier Polyline contains one or more off curve control points in between on curve points. Bezier curves can be filled to create curved shapes and are common in generalized Font representations.

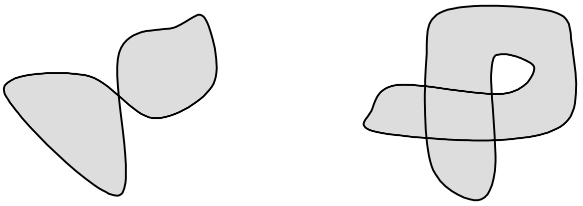
All line elements have direction from the start point to the end point.

Color fill may apply to a line. Refer to G.2.1.4.

G.2.1.4 Auto-Closure of A Line

When a line is specified with the Fill attribute, the line is considered as a closed line, which connects the start point and the end point using a straight line. The enclosed area of a closed line can be used for color fill.

The enclosed area is based on Nonzero fill rule. Following are two examples in which the light color indicates the enclosed area.



G.2.2 Polygon Elements

Polygon elements are closed representations of polyline, circular polyline and Bezier polyline elements. Polygons may have separate line and fill colors or may not be filled at all.

Polygon elements use the Nonzero fill rule for enclosed areas and can be used for color file.

A polygon element has its reference point at the starting point.

G.2.3 Simple Shape Elements

There are two types of simple shape elements: ellipse and rectangle.

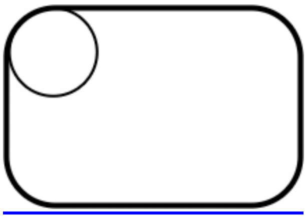
G.2.3.1 Ellipse

Ellipses are defined by their major axe, minor axe, center and angle of rotation. Circles are considered a special case of ellipse in which the major and minor axis are the same length.

G.2.3.2 Rectangle

Rectangles are represented by their center, length, height, and rotation angle. Squares are considered special rectangles in which the length and height are identical.

When the "round corner" indicator is set, the corner of the rectangle should be rounded. There is no specific radius of the round corner is defined. The recommended radius of the rounded corner should be 20% of the length of the shorter edge of the rectangle or the square.

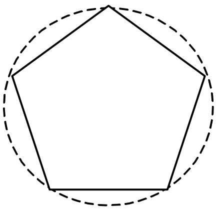


A simple shape element has its reference point at its center.

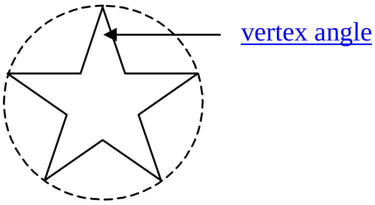
G.2.4 Special Shape Elements

There are 3 types of special shapes. Each shape has a reference point that determines its position. All special shapes except Grid have the reference point at its center. Shapes may have other parameters. These shapes include:

- **Regular Polygon:** a regular polygon has equal length of all its edges. In its original position, the bottom edge of the regular polygon should be aligned horizontally. A rotate angle can be optionally specified. Regular Polygon parameters include the number of vertex, the diameter of the reference circle and angle of rotation.



- **Star:** a star is defined by the number of corner vertex, the diameter of the reference circle, vertex angle and angle of rotation. In its original position, the bottom edge, which formed by two vertexes of the star, should be aligned horizontally. A rotate angle can be optionally specified. Vertex angles are predefined as 0, 36, 60, 90 degrees.



If the vertex angle is 0, a single line from center to vertex shall be drawn.

- **Grid:** a grid is a number of evenly distributed perpendicular lines. Its parameters include height, width, number of rows and columns (up to 16).

A special shape element has its reference point at its center.

G.2.5 Text Element

WVG supports text display inside the drawing. However it supports only the default font. To avoid inconsistency on different terminals, it is recommended to use vector based font. Text can be placed in a drawing with position, font size and rotate angle. Like other elements, text has attributes of line style, line color, line width. It can also be animated.

Control characters are ignored when the text is rendered except for the CR (Carriage Return). The CR indicates the text followed by should be displayed at the next line position. Multi-line text should be left aligned. There is no character spacing and line spacing defined. Recommended character spacing is 10% of the text height. Recommended line spacing is 20% of the text height.

When text encoding is GSM 7-bit, SMS character packing is used as defined in 3GPP TS 23.038 [9].

A text element has its reference point at top-left corner.

G.2.6 Group Elements

A set of elements can be grouped together.
A group of elements starts with a Group element, which is followed by a list of the elements in the group, and ends with an End_Group element. Two levels of grouping is allowed.

```
Group 1
Element 1
Element 2
……
          Group 2
          Element a
          Element b
          ……
          End Group
Element n
Element n+1
…..
End Group
```

A group element has its reference point at the reference point of its first element in the group.

G.2.7 Reuse Element

There are two usages of Reuse Element. The first is to repeat a set of elements in the bit-stream and the second is to display an element or group with a transform applied.

Repeat:

When the Encoder sees a set of elements that are identical to a previous set of elements, it replaces the latter set of elements with a Reuse element, so that encoded bit stream size will be minimized. A Reuse element uses the element_index and number_of_elements to repeat. When the Decoder see the Reuse Element, it will replace it with the set of elements that the Reuse element represents.

> NOTE: When calculating the element index of an element that follows a Reuse element in this case, the decoder should not count the Reuse element just as one element. Rather, the decoder should count the Reuse element it as the number of elements it represents. In other words, the index of elements after a Reuse Element in the bit-steam will be unchanged, so that another Reuse Element, which has an element_index, doesn't need to be changed after a Reuse replaces a number of elements.

One Reuse element can replace a maximum of 8 original elements.

Display:

Reuse Element can be used to display an element or a group of elements with a transform and/or changed attributes and/or display an array. Whether a Reuse Element references a group or a basic element depends on the element type that the element index in the Reuse Element points to. When reuse array is specified, the referenced element or group of elements is duplicated in rows and columns. The reference point of a reused array is at the center of the array.

G.2.8 Animation Elements

There are two types of animation elements, Simple Animation Element and Standard Animation Element. In the data format, an animation element is followed by another element or a group element that the animation applies to. An animation element cannot be followed by another animation element.

G.2.8.1 Simple Animation Elements

Simple animation is defined for WVG. All animation timing is based on an "Animation Cycle". WVG animation is repetitive. After completion of playing one cycle, a subsequent cycle play commences immediately.

There are two types of animation cycles defined, short cycle and long cycle. The time length of animation cycles are not defined. The time length of a long cycle should be twice the length of a short cycle. Recommended short cycle should collapses for 1 second and long cycle pay for 2 seconds.

There are two types of animations.

Visibility: an element can be visible or invisible during a specific cycle segments. A short cycle is divided into 4 time segments equally and a long cycle is divided into 8 time segments equally.

In the following example, a visibility for short cycle animation is defined. The element to be animated will blink following the pattern defined in the Visibility field below. Bit 1 indicates the element should be displayed during the time segment. Bit 0 indicates it should not be displayed during the time segment.

0	1	0	1

In the following example, a visibility for long cycle animation is defined. The element to be animated will blink following the pattern defined in the Visibility field below.

1	0	1	0	1	1	1	1

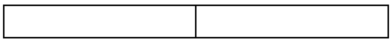
Transform: a start and an end transform can be applied to an element to describe the start and end position of a rotote, a scale, a translate animation or any combination of these action. When a transform element is omitted, it defaults to use the element's original position. An animation element must include at least one transform element. The animated element can also be a group to allow the animation action applied to a group of elements.

In Simple Animation, a transform from start position to end position should be completed in one cycle. A bounced flag can be turned on to allow "bouncing" animation. A bounced transform transforms the element from start position to end position in one cycle.

Transform in a short cycle:

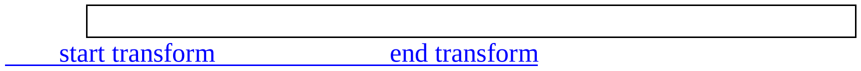
start transform end transform

Bounced Transform in a short cycle:

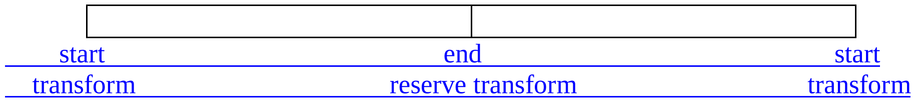


start end start
transform transform transform

Transform in a long cycle:



Bounced Transform in a long cycle:



Visibility and transform animation can be applied to the same element.

G.2.8.2 Standard Animation Element

An Standard Animation Element contains animation information such as begin transform position, end transform position, begin attribute, end attribute, begin time, end time, etc. This allows one animation element to represent a series of related images, which results in significant compression of the data stream. The WVG player interpolates between the beginning state and end state to achieve animation.

Animation Elements are not allowed inside Groups. Animation Rotation ranges from 0 to 360 degrees in both clockwise and counter-clockwise directions.

G.2.9 Frame Element

Frame element is as a marker of the start of a new frame. All elements before a Frame element belong to previous frame. The delay between two frames is defined as an infinite time interval. This means says that once a frame marker is reached, the elements that have been displayed on the screen at this time will stay on the screen until the user requests that the next frame should be displayed. The idea is that one can have multiple “pages” of graphics, such as a multi-page cartoon. The user can then study the first page and when finish can press a button (or trigger some other event) to see the next page of the cartoon. The mechanism of the user event is not defined and is left up to the application developer.

Here are parameters of a Frame Element:

- Keep last frame contents (or not). Zero means not keeping last frame contents, otherwise all the contents of previous frame will be kept.
- Fill in a new background color (or not). Zero means no new fill color is needed for this frame, otherwise a new background color will be used.
- New Background color.

A Frame element cannot appear in an element group. Reuse and Animation elements can not apply to a Frame element.

G.2.10 Local Element

This element defines the size and position of a local envelope.

The local envelope is a square area whose top-left corner is defined as the origin for its x and y-axis. The number of grid lines are pre-defined to 7, 15, 31 and 63. The resolution is constant in a local grid which is pre-defined at 1/27, 1/32, 1/48, 1/64, 1/85, 1/128, and 1/160 of the local envelope width. Actual envelope size can be determined by number of grid lines and grid resolution. The position of the local envelope is determined by the local envelope origin that falls at a coordinate within the global envelop.

A local element cannot appear in between another local start and local end element.

G.2.11 Extended Element

The Extended Element is defined to create objects which are not part of the base parsing level of defined objects in this specification and as a future proof method of expansion as defined by 3GPP technical committees. The Extended Element is intended for resolving problems in the current release. It may also be possible to use the Extended Element for potential enhancements in future releases. If the decoder encounters an Extended Element and the Extended Element

Type is unrecognized, it can gracefully skip this element by seeking past it in the bitstream, and continuing decoding at the next element in the bitstream.

An Extended Element contains the size of the Extended Element, the Extended Element type, and a series of bytes representing the payload data. The size field represents the payload data size in bytes. Note that when reading the payload data, bit alignment should be assumed (not byte alignment).

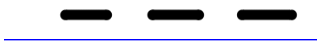
G.3 Element Attributes

Line, Polygon, Shape and Text elements can be applied with the following attributes.

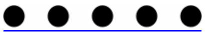
- Line width: 3 levels (fine, medium, thick). Default is fine.
- Line style: 4 types (solid, dash, dot and reserved). Default is solid.
- line color and fill color

Line Width
There are 4 line width settings defined, namely No Line, Fine, Medium and Thick. No specific width is defined for Fine, Medium and Thick. Recommended line widths are 1% or one pixel, 2% and 4% of the shorter dimension of the drawing. Line width for Fine, Medium and Thick should be at least 1 pixel. E.g., in a 120x80 pixel screen, the line width may appear as 1 pixel, 2 pixels and 3 pixels.

Line Type:
Dash Line: a dash line should start with a solid segment of the line. The length of the solid segments is recommended to be 4 to 6 times of the line width. The space between two solid segments is recommended to be 3-4 times of the line width.



Dotted Line: a dotted line is a string of circular dot on the path of a line. It is recommended that the diameter of rounded dot is same as the line width. The space between two dots shall be between 1 to 2 dot diameters.



Line Cap:
Line cap is Circular.

Line Joint:
Line joint is Round for line joint.

G.4 Element Transform

Transform element can be included in Group, Reuse and Animation elements and applied to line, polygon, shape, text and group elements. Supported transforms include rotate, translate and scale.

G.5 Character Size WVG Element

The Character Size WVG, or glyph is a subset variation of WVG. Character Size WVG uses a compact coordinate system with a half resolution global grid (7, 15, 31 and 63 grid lines), default color (monochrome), line elements (polyline, circular polyline and Bezier polyline) and a simplified drawing header.

G.6 Data Format BNF

The following notation is used in this document for BNF syntax:

< >	Enclose term names
|	Separates alternatives (exclusive OR)
[ ]	Square brackets enclose optional items in syntax descriptions.
{ }	{} Term enclosed is used zero or more times
()	() Enclose groups of alternative terms
...	From … to
;	Start with comments
0	Bit value 0 in bit stream
1	Bit value 1 in bit stream
‘ ‘	Terminator described by enclosed text

Notes for reading the BNF.

1. The bit value appearing at the left in the BNF indicates it is arranged in the front in the bit stream.
2. Notation 00…11 is equivalent to (00 | 01 | 10 | 11)
3. Notation (0 | 1 <val>) is used in the BNF in many occurrences for optionally omitting a value. In this example, it indicates either a specific value <val> can be used, or it can be omitted when default value can be used. The bit value 0 or 1 indicates if <val> is specified.

WVG (Wireless Vector Graphics)

```
<WVG> ::= ( 0 <character size WVG>)  | ( 1 <standard WVG> )

<character size WVG> ::= <character size WVG header> <line elements>

<standard WVG> ::= <standard WVG header> <elements>
```

Common

```
<text code mode> ::= 0 | 1                      ; 0 for 7-bit GSM character set. 1 for 16-bit UCS-2

<char> ::=      ‘unsigned 8 bit integer’        ; 7-bit GSM character value
                                                ; using GSM message packing into 8 bits
          |      unsigned 16 bit integer’       ; 16-bit UCS-2 value
                                         ; terminated by 0x00 or 0x0000. Control characters are prohibited.

<mask> :=       0 | 1                    ; 0 for false, 1 for true

<hint> :=       0 | 1                    ; 0 for false, 1 for true
```

Character Size WVG Header

```
<character size WVG header> ::=          0  ( <aspect ratio> <line element mask> <relative use>
                                                <parameters X-0> <parameters Y-0> )
                                                        ; standard header
                                 |       1 ( <line element mask> <relative use> <MaxXYInBits0> )
                                                        ; compact header. In this case, x and y grid are same,
                                                        ; default peak value 1.0, default aspect ratio1:1.
                                         ; Note: character size WVG always use compact coordinate mode

<line element mask> ::= <mask>           ; true for at least one polyline element in the drawing
                        <mask>           ; true for at least one circular polyline element in the drawing
                        <mask>           ; true for at least one Bezier polyline element in the drawing
```

```
<relative use> ::= 0 | 1                        ; 0 for all points use absolute coordinates,
                                                ; 1 for at least one point uses relative coordinate (offset mode)
<parameters X-0> ::= <MaxXInBits0> <peak description>
<parameters Y-0> ::= <MaxYInBits0> <peak description>
<MaxXInBits0> ::= <bits indicator >
<MaxXInBits0> ::= <bits indicator >
<MaxXYInBits0> ::= <bits indicator >
<bits indicator> ::= 00…11            ; 00 for 3 bits (max value 7), 01 for 4 bits (max value 15)
                                      ;10 for 5 bits (max value 31), 11 for 6 bits (max value 63)
<peak description> ::= 00…11                    ; 00:  peak value 1.0, no peak position required
                                                ; 01: peak value 1.5, peak position 0.5
                                                ; 10: peak value 1.5, peak position 0.3333
                                                ; 11: peak value 1.5, peak position 0.6667
```

Character Size WVG Elements

```
<line elements> ::= <line element> { <line element> }
<line element> ::= <line header>
              ( <polyline element> | <circular polyline element> |<Bezier polyline element> )
<line header> ::= <line element type> [ <point mode> ]           ; appear when <relative use> = 1
<line element type> ::=                         ; empty, when <line element mask> = 100, 010 or 100
                       0 | 1                    ; when <line element mask> = 011, 110, 110 or 101
                                                ; 0 for the firstelement with mask value 1 in the <line element mask>
                                                ; 1 for the second element with mask value 1 in <line element mask>
                       00..11                   ; 00 for polyline, 01 for circular polyline, 10 for Bezier polyline
                                                ; ( when <line element mask> = 111>)
<point mode> ::= 0 | (1 <offset bit use>)               ; 0 for use of absolute coordinate for <Next Point>
                                                        ; 1 for using relative coordinate (offset mode) for <Next Point>
```

Standard WVG Header

```
<standard WVG header> ::= <general info> <color configuration> <codec parameters> <animation settings>
<general info> ::= <version> 0 | ( 1 <text code mode> <author string> <title string> <time stamp> )
<version> ::= 0000…1111
<author string> ::= 0 | (1 <char> { <char> } )
<title string>  := 0 | (1 <char> { <char> } )
<time stamp> ::= 0 | (1 <year> <month> <day> <hour> <minute> <second> )
<year> ::='signed_13_bit_integer'
<month> ::='unsigned_4_bit_integer' ; range 1-12
<day> ::= 'unsigned_5_bit_integer' ; range 1-31
<hour>  ::= 'unsigned_5_bit_integer' ; range 0-23
<minute> ::= 'unsigned_6_bit_integer' ; range 0-59
<second> ::= 'unsigned_6_bit_integer' ; range 0-59>
```

Color

```
<color configuration> ::= <color scheme> <default colors>

<color scheme> ::=       00                                  ; black and white
                  |      010                                 ; 2-bit gray scale
                  |      011                                 ; 2-bit predefined color. 4 color value 00, 01, 10, 11 are
                                                             ; mapped to RGB color (0,0,0), (255,0,0), (0,255,0) and
                                                             ; (0,0,255) respectively
                  |      100                                 ; 6-bit RGB color
                  |      101                                 ; W3C websafe color
                  |      1100 <6-bit color palette>          ; 6-bit RGB color using 2nd color palette
                  |      1101 <8-bit color palette>          ; W3C websafe color using 2nd  palette
                  |      1110                                ; for 12 bits color mode
                  |      1111                                ; for 24 bits color mode

<6-bit color palette> ::= 00000…11111                        ; number of color. Maximum 32 color entries
                      {<6-bit RGB color>}                    ; specify color value from 0 to “number of color”-1

<8-bit color palette> ::= 0000000…1111111                    ; number of color. Maximum 128 color entries
                      { <8-bit websafe color> }              ;specify color value from 0 to “number of color”-1

                                          ; Note: the decoder will decide number of bits used by <indexed
                                          ; RGB/websafe color>  <indexed color> use 1 to 7 bits if <number of
                                          ; color> is  2, 3…4, 5…8, 9…16, 17…32, 33…64, 65…128.

<draw color> ::=         <b/w color>                         ; when color scheme is 00
                  |      <grayscale>                         ; when color scheme is 010
                  |      <2-bit predefined color>            ; when color scheme is 011
                  |      <6-bit RGB color>                   ; when color scheme is 100
                  |      <8-bit websafe color>               ; when color scheme is 101
                  |      <indexed RGB color>                 ; when color scheme is 1100
                  |      <indexed websafe color>             ; when color scheme is 1101
                  |      <12 bit RGB color>                  ; when color scheme is 1110
                  |      <24 bit RGB color>                  ; when color scheme is 1111

<b/w color> ::= 0 |      ; white
                  1      ; black

<grayscale> ::= 00…11  ; 00 for 24-bit RGB color (0,0,0),           01 for 24-bit RGB color (85,85,85)
                       ; 10 for 24-bit RGB color (170,170,170), 11 for 24-bit RGB color (255,255,255)

<2-bit predefined color> ::= 00…11        ;00 for 24-bit RGB color (0,0,0), 01 for 24-bit RGB color (255,0,0)
                                          ;10 for 24-bit RGB color (0,255,0), 11 for 24-bit RGB color (0,0,255)

<6-bit RGB color> ::= <2-bit R> <2-bit G> <2-bit B>

<indexed RGB color> ::= (0 | 1) | 00…11 | 000…111 | 0000…1111 | 00000…11111

                       ; map to 6-bit RGB color value defined in <6-bit color palette>

<8-bit websafe color> ::= 00000000…11111111

<indexed websafe color> ::= (0 | 1) | 00…11 | 000…111 | 0000…1111 |

                       00000…11111 | 000000…111111 | 0000000…1111111

                       ; map to 8-bit websafe color value defined in <8-bit color palette>


<2-bit R> ::= <2-bit color value>                    ; Red color value

<2-bit G> ::= <2-bit color value>                    ; green color value

<2-bit B> ::=  <2-bit color value>                   ; blue color value

<2-bit color value> ::= 00…11             ; 00, 01, 10 and 11  for color value 0, 85, 170 and 255
                                          ; defined in 0-255 color range respectively

<12-bit RGB color> ::= <4-bit R> <4-bit G> <4-bit B>       ;
```

```
<4-bit R> ::= <4-bit color value>                  ; Red color value
<4-bit G> ::= <4-bit color value>                  ; green color value
<4-bit B> ::=  <4-bit color value>                 ; blue color value
<4-bit color value> ::= 0000…1111                  ; left shift by 4 to convert to 24 bit color value
<24-bit RGB color> ::= <8-bit R> <8-bit G> <8-bit B>      ;
<8-bit R> ::= <8-bit color value>                  ; Red color value
<8-bit G> ::= <8-bit color value>                  ; green color value
<8-bit B> ::=  <8-bit color value>                 ; blue color value
<8-bit color value> ::= 00000000…1111111           ; intensity value of color value

<default colors> :=         ( 0 | (1 <default line color>))    ; use black  when first bit is 0
                            ( 0 | (1 <default fill color>))       ; use black  when first bit is 0
                            ( 0 | (1 <background color>))         ; use white  when first bit is 0
                            ;  If above color(s) are not
                            ; specified, use BLACK as <default line color> and <default fill color>, and use
                            ; WHITE as <background color>.

<default line color> ::= <draw color>
<default fill color> ::= <draw color>
<background color> ::= <draw color>
```

Codec Parameters

```
<codec parameters> ::= <element mask> <attribute mask> <generic parameters>
                            <coordinate parameters>
<coordinate parameters> ::=  ( 0 <flat coordinate parameters> )          ; flat coordinate mode
                            |  (1 <compact coordinate parameters> ) ; compact coordinate mode
<element mask> ::=          <mask>         ; true for at least one local envelop element in the drawing
                            <mask>         ; true for at least one polyline element in the drawing
                            <mask>         ; true for at least one circular polyline element in the drawing
                            <mask>         ; true for at least one Bezier polyline element in the drawing
                            <mask>         ; true for at least one simple shape element in the drawing
                            <mask>         ; true for at least one reuse element in the drawing
                            <mask>         ; true for at least one group element in the drawing
                            <mask>         ; true for at least one animation element in the drawing
                            (0 | (1        ; extension bit. 1 for rare masks are followed by
                                  <mask>         ; true for at least one polygon element in the drawing
                                  <mask>         ; true for at least one special shape element in the drawing
                                  <mask>         ; true for at least one frame element in the drawing
                                  <mask>         ; true for at least one text element in the drawing
                                  <mask>         ; true for at least one extended element in the drawing
                                  <mask>         ; reserved
                                  <mask>         ; reserved
                            )     ;The decoder should decide how many bits to be used by <element type>
                            )     ; according to number of "1"s in the <element mask>. Number of bits
                                  ; used by <element type> can be 0 (if only one "1" in <element mask>),
                                  ; 1 (if 2 "1"s), 2 (if  3 or 4 "1"s), 3 (if 5-8 "1"s) or 4(if more than 8
                                  ; "1"s). Value of <element type> that is to represent a specific element
                                  ;  type is same as the order of  the specific mask  in the <element mask>
                                  ; that represents this type of element. For example, if <element mask> is
                                  ; 1100000000010000, <element  type> will use 2 bits and value 00, 01, 10
                                  ;  (11 is not used) represent circular polyline, rectangle and animation
```

```
                                    ;  elements respectively.
<attribute masks> ::= <line type mask> <line width mask> <line color mask> <fill mask>
<line type mask> ::= <mask>             ; true when at least one element uses line type attribute
<line width mask> ::= <mask>            ; true when at least one element uses line width attribute
<fill mask> ::= <mask>                  ; true when at least one element uses fill attribute
<line color mask> ::= <mask>            ; true when at least one element uses line color
```

Generic Parameters

```
<generic parameters> ::= (0 | (1 <angle resolution> <angle in bits> )      ; 0 for default (22.5 degree, 3 bits)
                           (0 | (1 <scale resolution> <scale in bits> )    ; 0 for default (1/4, 3 bits)
                           (0 | (1 <index in bits>) ; 0 for default (both  3 bits)
                         [<curve offset in bits> ]
                         ; <curve offset in bits> appear when <mask> for <circular polyline element>
                         ; or <polygon element> is true

<angle resolution> ::= 00…11            ; 00  for angle unit is 1  degree; 01 for angle unit is 5.625 degree
                                        ; 01  for angle unit is 11.25 degree; 11 for angle unit is 22.5 degree

<angle in bits> ::= 000…111             ; number of bits used by <angle value> is from 2 to 9 bits

<angle value> ::= ‘unsigned angleInBits+2-bit integer’
                                                      ; angle unit is decided by <angle resolution>

<scale resolution> ::= 00..11           ; 00 for 1/4 as scale unit.  01 for 1/16 as scale unit
                                        ; 10  for 1/64 as scale unit; 11 for 1/256 as scale unit

<scale in bits> ::= 000…111             ; number of bits used by <scale value> is from 3 to 10 bits

<scale value> ::=  ‘signed scaleInBits+3-bits integer’
                                                      ; scale unit is decided by <scale resolution>
                                                      ; scale value include a sign bit

<index in bits> ::= 000…111             ; number of bits used by <index> are from 3 to 10 bits

<index> ::= <index value>

<index value> ::= ‘unsigned IndexInBits+3-bit integer’

<curve offset in bits> ::= 0 | 1        ; 0 for using 4 bits (15 levels)
                                        ; 1 for using 5 bits (31 levels)
```

Compact Coordinate Parameters

```
<compact coordinate parameters> ::= <aspect ratio> <TransXYInBits1>   ; 0 for default aspect ratio 1:1
<parameters X-1> <parameters Y-1> <redefine resolution hint>

<aspect ratio> ::=    00 |                            ; aspect ratio = 1:1
                 ( (  01          ; aspect ratio = 4:3
                    | 10          ; aspect ratio = 16:9
                    | 1100                            ; aspect ratio = 64:27
                    | 1101                            ; aspect ration = 256:81
                    | 1110                            ; aspect ration = 1024:243
                    | 1111                            ; aspect ration = 4096:729
                   ) [ <display orientation> ]        ; <display orientation> appears when standard WVG
                 )                                    ; character size WVG uses landscape as default

<display orientation > ::= 0 | 1                ; 0 for landscape, 1 for portrait

<parameters X-1> ::= <MaxXInBits1> <coordinate parameters>

<parameters Y-1> ::= <MaxYInBits1> <coordinate parameters>

<coordinate parameters> ::= <peak value> <peak position> <peak width>

<MaxXInBits1> ::= 00…11                         ; the number of grid lines
```

```
<MaxYInBits1> ::= 00…11                    ; the number of grid lines
                                           ; 00 for 15, 01 for 31,10 for 63, 11 for 127
<peak value> ::= 00…11                     ; 00 for1.0, 01 for 1.5, 10  for 2.0, 11 for 2.5
                                           ; if , <peak value> is 00, <peak position>,
                                           ; <peak width> and <transition width> are ignored.
<peak position> ::=    0000…1100           ; 0-12. Peak position = value/12 from envelope left.
                  |    1101                ; reserved
                  |    1110                ; reserved
                  |    1111                ; reserved
<peak width> ::= 00…11                     ; 00 for 0.3, 01 for 0.4, 10 for 0.5, 11 for 0.6
                  ;<peak width> value are to the scale of total global envelope width.
<redefine resolution hint> ::= <hint>      ; true when at least one element uses 'redefine resolution'attribute
<TransXYInBits1> ::= 00..11       ; number of bits to encode translation and center of transform
                                  ; 00 for 5 bits, 01 for 6 bits, 10 for 7 bits, 11 for 8 bits
```

Flat Coordinate Parameters

```
<flat coordinate parameters> ::= <drawing width> ( 0 | 1 (<drawing height>))       ; 0 means height = width
                        <MaxXInBits2><MaxYInBits2> < XYAllPositive> <TransXYInBits2>
                        <OffsetXInBitsLevel1> <OffsetYInBitsLevel1>
                        <OffsetXInBitsLevel2> <OffsetYInBitsLevel2>
<drawing width> ::= 'unsigned 16-bit integer'
<drawing height> ::= 'unsigned 16-bit integer'
<MaxXInBits2> ::= 'unsigned_4_bit_integer'
                        ; number of bits to encode X coordination
<MaxYInBits2> ::= 'unsigned_4_bit_integer'
                        ; number of bits to encode Y coordination
<XYAllPositive> ::= "unsigned_1_bit_integer'
                        ; 0 means not all x/y are positive
                        ; 1 means all x/y are positive
<TransXYInBits2> ::= 'unsigned_4_bit_integer'    ; number of bits to encode translation and center of transform
<OffsetXInBitsLevel1> ::= 'unsigned_4_bit_integer'
<OffsetYInBitsLevel1> ::= 'unsigned_4_bit_integer'
<OffsetXInBitsLevel2> ::= 'unsigned_4_bit_integer'
<OffsetYInBitsLevel2> ::= 'unsigned_4_bit_integer'
<NumPointsInBits> ::= 'unsigned_4_bit_integer'
```

Animation Settings

```
<animation settings> ::= [ <animation mode> ]          ;appear when <animation element> exist
                  [ <frame timing> ]                   ; appear when <frame element> exist
<animation mode> ::= 0 | 1        ; 0 for simple animation;  1 for standard animation
<frame timing> ::= 0 | (1         ; 0 means infinite delay between frames,
                                  ; 1 means reserved
```

Element

<elements> := <element> { <element> }

<element> := <element type> (<basic element> |
<frame element> | <group element> | <re-use element> |
<animation element> | <extended element> | <local envelop element>)

<element type> ::= | 0…1 | 00..11 | 000…111 | 0000…1111 *; empty is allowed*

; decided by <element mask>. Please refer to <element mask>

<animation element> := <simple animation element> | <standard animation element>

; if <animation mode> is 0, all animation elements in the drawing are <simple animation element>
; if <animation mode> is 1, all animation elements in the drawing are <standard animation element>

<basic element>::= <basic element header> (<polyline element> | <circular polyline element>

| <Bezier polyline element> | <polygon element> | <simple shape element>

| <special shape element> | <text element>)

Basic Element Header

<basic element header> ::= [<resolution redefinition>] *; appear when using global coordinates*
; in compact coordinate mode
(0 | (1<offset bit use>))
; specify measurement mode for <Next Point> <width> <height> and <diameter>
; etc. the 0|1 indicator only exist in compact coordinate mode
; in compact coordinate mode, 0 for absolute mode, 1 for offset mode
[0 | (1<attributes set>)] *; appears when <attribute masks> does not equal*
; to 0000
; 0 for *using default attributes defined in <drawing header>*
; 1 for *using the following specific attributes*

<Offset Bit Use> ::= <Offset X Use><Offset Y Use>

<Offset X Use> ::= 0 | 1

; when in compact coordinate mode, 0 means offset X will use 3 bits.,

1 means use 4 bits

; when in flat coordinate mode, 0 means offset X will use <OffsetXInBitsLevel1>,

1 means use <OffsetXInBitsLevel2>

<Offset Y Use> ::= 0 | 1

; when in compact coordinate mode, 0 means offset X will use 3 bits,

1 means use 4 bits

; when in flat coordinate mode, 0 means offset X will use <OffsetYInBitsLevel1>,

1 means use <OffsetYInBitsLevel2>

<resolution redefinition>::= ; *; empty, do not redefine resolution*
; when <redefine resolution hint> is false or in local scope

| 0 *; do not redefine resolution*
; when <redefine resolution hint> is true and in global scope

| (1 <coordinate resolution>) *; redefine resolution*
; when <redefine resolution hint> is true and in global scope

<coordinate resolution> ::= 000…111 *; decide the grid line interval by a scale of width*
; or height of the global envelope whichever is short.
; 0-7 for 1/27, 1/32, 1/38, 1/48, 1/64,
; 1/85, 1/128 and 1/160 respectively

```
                                        ; after definition, the element still use <MaxXInBits1>,
                                        ; <MaxYInBits1>, <MaxYInBits2>,<MaxXInBits2>,
                                        <MaxXInBits1> unless it uses offset mode
```

Element Attributes

```
<attribute set> ::=     [ <line type> ]          ; appear when <line type  mask> is true
                        [ <line width> ]         ; appear when <line width mask> is true
                        [ 0 | (1 <line color>) ] ; appear when <line color mask> is true and
                                                 ;  <line width> is not zero
                        [ 0 | (1                 ; 0 for no fill; 1 for with fill
                             (0 | (1 <fill color>)) ; use <default fill color> if <fill color> absent
                             )                   ; appear when <fill mask> is true
                        ]                  ; Note: line type and fill are not used by <text element> but still exist here,

<line width> ::= 00…11                     ; 00 for no line, 01 for Fine, 10 for medium, 11 for thick
                                           ; 00 is only valid when <fill color> is specified

<line type> ::= 00…11                      ; 0 for solid, 1 for dash line, 2 for dotted line

 <fill color > ::= <draw color>

<line color> ::= <draw color>
```

Transform

Note: signed integers use Two’s Complement representation.

```
<Transform> ::= ( (0 <point> ) | (1<TranslateX><TranslateY>) )   ; mandatory new position using two ways

                0 | (1                                          ; optional other transforms<Angle>
     <ScaleX><ScaleY> < CX>< CY>)

                ; Default rotation and scale center of <basic element> is the first point of lines, center of rectangle,
                ; ellipse and special shapes. Default rotation and scale center of <group element> is the rotation and
                ; scale center of the first basic element in the group.

<Angle> ::= 0 | (1 <Angle Value> )                   ; 0 means angle will use default value which is 0

<TranslateX> ::= 0 | (1 <TranslateX Value> )         ; 0 means translate x  will use default value which is 0

<TranslateX Value> ::= ’signed_TransXYInBits2_bit integer’        ; when in flat coordinate mode

                | ‘signed TransXYInBits1+4 integer’               ; when in compact coordinate mode

<TranslateY> ::= 0 | (1 <TranslateY Value> )         ; 0 means translate y will use default value which is 0

<TranslateY Value> ::= ’signed_TransXYInBits2_bit integer’        ; when in flat coordinate mode

                | ‘signed TransXYInBits1+4-bit integer’           ; when in compact coordinate mode

<ScaleX> ::= 0 | (1<Scale value> )                   ; 0 means scale will use default value which is 1.0
<ScaleY>::= 0 | (1 <Scale value> )                   ; 0 means scale will use default value which is same as
                                                     ; absolute value of <ScaleX>

<CX> ::= 0 | (1 <CX value> )               ; translation of rotation and scale center; 0 means it will use default

                                           ; value which is 0

<CX value> ::= ’signed_TransXYInBits2_bit integer’                ; when in flat coordinate mode

                | ‘signed TransXYInBits1+4-bit integer’           ; when in compact coordinate mode

<CY> ::= 0 | (1 <CY value> )               ; 0 means it will use default value which is 0

<CY value> ::= ’signed_TransXYInBits2_bit integer’                ; when in flat coordinate mode

                | ‘signed TransXYInBits1+5-bit integer’           ; when in compact coordinate mode
```

Polyline Element

```
<polyline element> ::= [ <numberOfPoints> ] <First Point> { <Next Point> } [ <point terminator> ]
```

```
                              ; specifies a start point, zero or many intermediate points and an end point.
                              ; <numberOfPoints> appears only when in flat coordinate mode
                              ; <point terminator> appears only when in compact coordinate mode

<point terminator> ::=        111…111111              ; Absolute mode in character size WVG. Same number of
                                                      ; bits of <MaxXInBits0> or <MaxXYInBits0>
                    |         1111…1111111            ; Absolute mode in standard WVG. Same number of bits of
                                                      ; <MaxXInBits1>or <MaxLocalXYInBits>
                                             |        100  | 1000        ; Offset mode (relative).
```

Circular Polyline Element

Note: signed integers use Two's Complement representation.

```
<circular polyline element> ::=    <curve hint> [ <numberOfPoints> ] <point> <curve offset> <point>
                              { <curve offset> <point> } [ <offset terminator> ]

                              ; <numberOfPoints> appears only when use
                              ; flat coordinate  mode
                              ; <offset terminator> appears only when use

                              ; compact coordinate mode

<curve hint> ::= <hint>

<curve offset> ::=            ( 0 | (1 <curve offset value>) )         ; when <curve hint> is true
                    |         <curve offset value>                      ; when <curve hint> is false

<offset value> ::=            'signed 4-bit integer'          ; when <curve offset in bits> = 0
                                                              ; or in character size WVG
                    |         'signed 5-bit integer'          ; when <curve offset in bits> = 1
                              ; Curve offset ratio r = e/L
                              ; Where e is actual curve offset(can be positive or negative),
                              ; L is distance between adjacent nodes
                              ; We use a signed integer value v to represent. v = round(r*k);
                              ; Where k = 2^n - 2  (n is number of bits used for <offset value>)

<offset terminator> ::=       ( 1 <curve offset bits>)        ; when <curve hint> is true
                    |         < curve offset bits >                    ; when <curve hint> is false

<curve offset bits> ::=                 1000                  ; when <offset in bits> = 0
                              |         10000                 ; when <offset in bits> = 1
```

Bezier Polyline Element

```
<Bezier polyline element> ::= [ <NumberOfPoints> ]

                              <First Point> {[<OnCurve>] <Next Point>} [ 1 <point terminator>]

                    ; Same data format for PolyBezCurve, and PolygonBezCurve

                    ; <numberOfPoints> appears only when in flat coordinate moed

                    ; "1 <point terminator>" appears only when in compact coordinate mode

<NumberOfPoints> ::= 'unsigned_NumberOfPointsInBits_bit integer'

<OnCurve> ::= 0 | 1

                              ; 0 – off curve

                              ; 1 – on curve
```

Polygon Element

Polygon element is actually a closed polyline (including circular and Bezier polyline)

```
<polygon element> ::= ( 00 <polyline element> ) | (01 <circular polyline element> )

                               | (10 <Bezier polyline element> )
```

Simple Shape Element

```
<simple shape element> ::= (0 <rectangle element> ) | (1 <ellipse element> )

<rectangle element>::=<Point><Width><Height><rounded flag> (0 | (1 <Angle>))

<ellipse element>::=<Point><Width><Height> (0 | (1 <Angle>))

<Width> ::= <X> | <Offset X>                ; decided by measurement mode (see <basic element header>)

<Height>::= 0 | (1 <HeightValue> )           ; 0 means the height is same as width, height will not be encoded

<HeightValue> ::= <Offset Y> | <Y>          ; decided by measurement mode (see <basic element header>)

<rounded flag> ::=       0 | 1              ; 0 for straight corner, 1 for rounded corner
```

Special Shape Element

```
<special shape element> ::=     <point>
                          00 ( <vertex> < diameter > (0 | ( (1 <angle>)  )                    ; regular polygon
                    |     01 ( <vertex> <vertex angle> < diameter > (0 | (1 <angle> ) )       ; star
                    |     10 ( <rectangle size> <rows> <columns> )                            ; grid
                    |     11                      ; reserved

<diameter > ::= <X> | <Offset X>            ; diameter of circle or vertex

<rectangle size>::= <width> <height>

<vertex> ::= 000…111                        ; number of vertex = <vertex> + 3

<vertex angle> ::= 00…11                    ; 00 for 0 degree, 01 for 36 degree
                                            ; 10 for 60 degree, 11 for 90 degree

<rows> ::= 0000…1111                        ; rows = <rows> + 1

<columns> ::= 0000…1111                     ; columns = <columns> + 1
```

Text Element

```
<text element> ::= <point> <font size> <angle> <text code mode> { <char> }

                                            ; <point> is top-left  corner of the text.

<font size> ::= <Y> | <Offset Y>
```

Local Envelop Element

```
<local envelop element> ::= ( 0 <local envelope description> <point> )
                          ; local start
                          ; <point> is top- left corner of the local envelope in global coordinates.
                          ; Elements in the local envelope scope use local coordinates and measurements

                          |    1      ; local end


<local envelope description> ::= <direction> <coordinate resolution> <MaxLocalXYInBits>


<direction> ::=   00 |      ; x and y axis are at same direction of the global envelop
                  01 |      ; x axis is at negative direction of x axis of the global envelop, and y at same direction
                  10 |      ; x and y axis are at negative direction of the global envelop
                  11        ; y axis is at negative direction of y axis of the global envelop, and x at same direction

<MaxLocalXYInBits> ::= 00…11                ; 00 for 3 bits(max value 7), 01 for 4 bits (max value 15),

                                             10 for 5 bits (max value 31), 11 for 6 bits (max value 63)
```

Group Element

```
<group element> ::=      (0 (0 | (1 <transform>) ) <display> )        ; start of group. Transform is optional
```

```
          |      1                                  ; end of group

<display> ::=  0 | 1                            ; 0 – no display when render; 1 – display when render
```

Re-use Element

```
<re-use element> ::=  <element index>          ; point to the element to be re-used
                                               ; only <basic element>,<group element> and
                                               ; <re-use element> can be reused
               ( 0 <number of elements> )          ; simple repeat (usually used in multi-frame cases)
          |    ( 1                                 ; re-use with changes
                    <transform>
                                                   ; re-use with transformation
                    0 | (1 <array parameter>)      ; array. It should be performed as the last step
                    0 | (1 <OverrideAttributeSet> )
               )
<element index> ::= <index value>       ; the element sequence number in whole drawing
<number of elements>::='unsigned 3-bits integer'       ; number of elements will be repeated when
encode
<array parameter> ::= <rows> [ <height> ] <columns> [ <width> ]
               ; <height> indicates whole height of the array, appears when <rows> is non-zero
               ; <width> indicates whole width of the array, appears when <colunms> is non-zero
<OverrideAttributeSet> ::= <AttributeSet>                   ; override attributes
```

Frame Element

```
<Frame> ::= <KeepLastFrameContentFlag><HasFilledColorFlag>[<Filled Color>]
<KeepLastFrameContentFlag>::='unsigned 1-bit integer'
                              ;  keep the image of the last frame on the screen, or clear it
                              ; value 0 - Do not keep last frame content.
                              ; value 1 - Keep last frame content.
<HasFilledColorFlag> ::='unsigned 1-bit integer'
                              ; value 0 - no filled color
                              ; value 1 - has filled color
<Filled Color>::=<draw color>
                              ; new background color for the frame
```

Simple Animation Element

```
<simple animation element> ::=  <cycle type>
               ( 0 | ( 1 <visibility parameter> )
               ( 0 | ( 1 <transform> )             ; begin transform
               ( 0 | ( 1 <transform> )             ; end transform
               ( 0 | 1 )                           ; 0 for no bouncing.  1 for bouncing
               ( 0 | 1 <rotation direction> )      ;0  for no rotation or specified by <transform>.
                                                   ; 1 for round rotation and will override angles defined in
                                                    ; <transform>
               ; all animation actions use reference point of the animated <basic element> being reused
               ; or the  reference point of the first element in the animated <group element>
<cycle type> ::= 0 | 1            ; 0 indicates short animation cycle; 1 indicates long animation cycle
<visibility parameter> ::= <visibility timing>
<visibility timing> ::= 0000…1111 | 00000000…11111111
     ; One blinking cycle is  divided into four equal time steps for short
```

; animation cycle or eight steps for long animation cycle. <visibility timing> is a map of time steps in
; which 0 represents invisible and 1 represents visible. Note that in above map, consequence time steps
; is from left to right, or from first order to later order in bit stream.

<rotation direction> ::= 0 | 1 *; 1 for clockwise rotating. 0 for counter-clockwise rotating*

Standard Animation Element

<standard animation element>::= <element index> <BeginTransform><EndTransform><Rotation Direction>
<Round><Begin Attribute><EndAttribute><BeginTime><Duration><ExistAfter>

<BeginTransform> ::= 0 | (1<Transform>)

;0 – means use (start from) default transform:
; Angle=0, TranslateX=0, TranslateY=0, ScaleX=256, ScaleY=256, Cx=0, Cy=0
;1 – means Transform follows

<EndTransform> ::= 0 | (1 <Transform>)

;0 – means use (end at) default transform
; Angle=0, TranslateX=0, TranslateY=0, ScaleX=256, ScaleY=256, Cx=0, Cy=0

;1 – means Transform follows

<Rotation Direction> ::= 0 | 1 *;0 – counter clockwise*

;1 – clockwise

<Round> ::= 0 | 1 *;0 – rotate 360 degrees*

;1 – no rotation

<BeginAttribute> ::= 0 | (1 <Attribute Set>)

;0 – use default attribute set (starts from current attribute set)

;1 – Attribute Set follows

<EndAttribute> ::= 0 | (1 <Attribute Set>)

;0 – use default attribute set (ends at the current attribute set)

;1 –Attribute Set follows

<BeginTime> ::= 'unsigned 12-bit integer'

<Duration> ::= 'unsigned 12-bit integer'

<ExistAfter> :: = 0 | 1 *; 0 – animation element will disappear after the animation is finished*

; 1 – animation element will persist after the animation is finished

Extended Element

<Extended> ::= <SizeOfSize><Size><ExtendedElementType>{<payload>}

<SizeOfSize>::='unsigned_5_bit integer'

; the bit size of the Size field

<Size>::='unsigned-<SizeOfSize>-bit integer'

; size of extended element data after ExtendedElementType, in bytes

<ExtendedElementType>::='unsigned_8_bit integer'

; element type of extended element

<payload>::='unsigned_8_bit integer'

; encoded extended element data. The size should be the same as the Size field of Extended, above.

Position and Measurement

Note: signed integers use Two's Complement representation.

```
<First Point>::=<point>                     ; first point of a polyline or polygon (including circular and
                                            ; Bezier polygons)
<Next Point> ::= <point> |                  ; when use absolute mode
                 <Offset>                   ; when use offset mode
<point> ::= <X> <Y>
<X> ::= ’signed MaxXInBits2-bit integer’  ; when in flat coordinate mode and <XYAllPositive> = 0
        | ‘unsigned MaxXInBits2-bit integer’             ; when in flat coordinate mode and <XYAllPositive> = 1
        | ‘unsigned MaxXInBits1+4-bit integer’           ; when in compact coordinate mode and in global scope
        | ‘unsigned MaxLocalXYInBits+4-bit integer’      ; when in compact coordinate mode and in local scope
        | ‘unsigned MaxXInBits0+3-bit integer’           ; when in character size WVG (use standard header)
        | ‘unsigned MaxXYBits0+3-bit integer’            ; when in character size WVG (use compact header)
<Y> ::= ’signed MaxYInBits2-bit integer’  ; when in flat coordinate mode and <XYAllPositive> = 0
        | ’unsigned MaxYInBits2-bit integer’             ; when in flat coordinate mode and <XYAllPositive> = 1
        | ‘unsigned MaxYInBits1+4-bit integer’           ; when in compact coordinate mode and in global scope
        | ‘unsigned MaxLocalXYInBits+4-bit integer’      ; when in compact coordinate mode and in local scope
        | ‘unsigned MaxYInBits0+3-bit integer’           ; when in character size WVG (use standard header)
        | ‘unsigned MaxXYBits0+3-bit integer’            ; when in character size WVG (use compact header)
        ; Note: in compact coordinate mode,<X> and <Y> do not use the maximum number of the unsigned integer
<Offset> ::= <Offset X> <Offset Y>
<Offset X> ::= <signed offset X>                         ; when used by <Next Point>
               | <unsigned offset X>                     ; when used in other cases
<signed offset X> = ’signed OffsetXInBitsLevel1-bit integer’
                                            ;when in flat coordinate mode and <offset bit use> = 0
               | ’signed OffsetYInBitsLevel2-bit integer’
                                            ;when in flat coordinate mode and <offset bit use> = 1
               | ‘signed 3-bit integer’     ;when in compact coordinate mode and <offset bit use> = 0
               | ‘signed 4-bit integer’     ;when in compact coordinate mode and <offset bit use> = 0
<unsigned offset Y> ::= ’unsigned OffsetYInBitsLevel1-bit integer’
                                            ;when in flat coordinate mode and <offset bit use> = 0
               | ’unsigned OffsetYInBitsLevel2-bit integer’
                                            ;when in flat coordinate mode and <offset bit use> = 1
               | ‘unsigned 3-bit integer’   ;when in compact coordinate mode and <offset bit use> = 0
               | ‘unsigned 4-bit integer’   ;when in compact coordinate mode and <offset bit use> = 1
```