Final draft ETSI EN 300 812 V2.1.1 (2001-08)

European Standard (Telecommunications series)

Terrestrial Trunked Radio (TETRA); Security aspects; Subscriber Identity Module to Mobile Equipment (SIM-ME) interface

Reference

REN/TETRA-07043

Keywords

card, security, SIM, TETRA

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Content

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://www.etsi.org/ipr).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This European Standard (Telecommunications series) has been produced by ETSI Project Terrestrial Trunked Radio (TETRA), and is now submitted for the ETSI standards One-step Approval Procedure.

Proposed national transposition dates	
Date of latest announcement of this EN (doa):	3 months after ETSI publication
Date of latest publication of new National Standard or endorsement of this EN (dop/e):	6 months after doa
Date of withdrawal of any conflicting National Standard (dow):	6 months after doa

1 Scope

The present document defines the interface between the Subscriber Identity Module (SIM) and the Mobile Equipment (ME) for use during the network operation phase of TETRA as well as those aspects of the internal organization of the SIM which are related to the network operation phase. This is to ensure interoperability between a SIM and a ME independently of the respective manufacturers and operators. The concept of a split of the MS into these elements as well as the distinction between the TETRA network operation phase, which is also called TETRA operations, and the administrative management phase is described in the User Requirement Specification ETR 295 [6].

The present document defines:

- the requirements for the physical characteristics of the SIM, the electrical signals and the transmission protocols;
- the model which shall be used as a basis for the design of the logical structure of the SIM;
- the security features; This edition of the standard covers the security mechanisms for ITSI based services including authentication and OTAR for keys addressed to an ITSI;
- the interface functions;
- the commands;
- the contents of the files required for the TETRA application;
- the application protocol.

The present document does not specify any aspects related to the administrative management phase. Any internal technical realization of either the SIM or the ME are only specified where these reflect over the interface. The present document does not specify any of the security algorithms which may be used.

The physical SIM described in the present document is a removable Integrated Circuit (IC) card. The SIM is an optional device within TETRA MSs. The present document does not preclude the implementation of fully functional MSs without a SIM. All references to mobile equipment in the present document are to be taken to mean mobile equipment which have been designed to operate with a SIM.

The present document deals with all aspects of trunked mode MS operation. For direct mode MS operation key user operation is supported by the SIM but not key holder or key generator operation. Furthermore, storage of information for direct mode MS operation in repeater and gateway mode are supported, but any extra storage required in the direct mode repeater or direct mode gateway terminals themselves is not supported.

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 and/or edition number or version number) or non-specific.
- For a specific reference, subsequent revisions do not apply.
- For a non-specific reference, the latest version applies.

[1] ETSI ETS 300 392-1: "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 1: General network design".

[2] ETSI EN 300 392-2: "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 2: Air Interface (AI)".

[3] ETSI EN 300 392-7: "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 7: Security".

[4] ETSI EN 300 392-12-22: "Terrestrial Trunked Radio (TETRA); Voice plus Data (V+D); Part 12: Supplementary services stage 3; Sub-part 22: Dynamic Group Number Assignment (DGNA)".

[5] ETSI ETS 300 394-2: "Terrestrial Trunked Radio (TETRA); Conformance testing specification; Part 2: Protocol testing specification for Voice plus Data (V+D)".

[6] ETSI ETR 295: "Terrestrial Trunked Radio (TETRA); User requirements for Subscriber Identity Module (SIM)".

[7] ETSI ETS 300 396-6: "Terrestrial Trunked Radio (TETRA); Direct Mode Operation (DMO); Part 6: Security".

[8] ETSI ETS 300 812 Edition 1: "Terrestrial Trunked Radio (TETRA); Security aspects; Subscriber Identity Module to Mobile Equipment (SIM - ME) interface".

[9] ETSI TS 100 977: "Digital cellular telecommunications system (Phase 2+) (GSM); Specification of the Subscriber Identity Module - Mobile Equipment (SIM - ME) interface" (GSM 11.11).

[10] ETSI TS 100 900: "Digital cellular telecommunications system (Phase 2+) (GSM); Alphabets and language-specific information "(GSM 03.38).

[11] ETSI TS 100 906: "Digital cellular telecommunications system (Phase 2+) (GSM); Mobile Stations (MS) features"(GSM 02.07).

[12] ETSI TS 100 907: "Digital cellular telecommunications system (Phase 2+) (GSM); Man-Machine Interface (MMI) of the Mobile Station (MS)"(GSM 02.30).

[13] ETSI TS 100 927: "Digital cellular telecommunications system (Phase 2+) (GSM); Numbering, addressing and identification"(GSM 03.03).

[14] GTS GSM 04.08: "Digital cellular telecommunications system (Phase 2+) (GSM); Mobile radio interface layer 3 specification".

[15] GTS GSM 11.12: "Digital cellular telecommunications system (Phase 2) (GSM); Specification of the 3 Volt Subscriber Identity Module - Mobile Equipment (SIM - ME) interface ".

[16] ISO 7810 (1995): "Identification cards - Physical characteristics".

[17] ISO/IEC 7811-1 (1995): "Identification cards - Recording technique - Part 1: Embossing".

[18] ISO/IEC 7811-3 (1995): "Identification cards - Recording technique - Part 3: Location of embossed characters on ID-1 cards".

[19] ISO/IEC 7816-1 (1998): "Identification cards - Integrated circuit(s) cards with contacts - Part 1: Physical characteristics".

[20] ISO/ISO 7816-2 (1999): "Information technology - Identification cards - Integrated circuit(s) cards with contacts, Part 2: Dimensions and location of the contacts".

[21] ISO/IEC 7816-3 (1997): "Information technology - Identification cards - Integrated circuit(s) cards with contacts - Part 3: Electronic signals and transmission protocols".

[22] ISO/IEC 7816-5: "Identification cards - Integrated circuit(s) cards with contacts - Part 5: Numbering system and registration procedure for application identifiers".

[23] ISO 639 (1988): "Code for the representation of names of languages".

[24] ISO/IEC 8859-1 (1998): "Information technology - 8 bit-single byte coded graphic character sets - Part 1: Latin alphabet No. 1".

[25] ENV 1375-1: "Identification card systems - Intersector integrated circuit(s) card additional formats - Part 1: ID-000 card size and physical characteristics".

[26] ITU-T Recommendation T.50: "International Reference Alphabet (IRA) (Formerly International Alphabet No. 5 or IA5) – Information technology – 7-bit coded character set for information interchange".

[27] ITU-T Recommendation E.118: "The international telecommunication charge card".

[28] ITU-T Recommendation E.164: "The international public telecommunication numbering plan".

3 Definitions, symbols and abbreviations

3.1 Definitions

For the purposes of the present document, the terms and definitions given in ETS 300 392-1 and the following apply:

access conditions: set of security attributes associated with access to an Elementary File (EF):

- ADM (administrative):

  indicates an access condition defined by the card issuer. Before issue of the card ADM serves as a placeholder for an access condition to be defined by the card issuer. Any access condition may be assigned. The assigned access condition is used during the usage phase of the SIM;

- AUTI (authorized immediate):

  defines access conditions to an EF under which access shall is only possible immediately following successful authentication of the Switching and Management Infrastructure (SwMI);

- CHVn (card holder verification):

  defines the access condition to an EF which requires verification of the user identity (n = 1 or n = 2);

- NEV (never):

  access to the EF is never allowed across the SIM-ME interface.

administrative phase: part of the card life between the manufacturing phase and the usage phase

application: set of security mechanisms, files, data and protocols (excluding transmission protocols)

application protocol: set of procedures required by the application which are located and used in the Integrated Circuit (IC) card and outside the IC card (external application)

card holder verification: authentication of the user to the SIM card

card session: link between the card and the external world starting with the Answer To Reset (ATR) and ending with a subsequent reset or a deactivation of the card

current directory: latest Master File (MF) or Dedicated File (DF) selected

current Elementary File (EF): latest EF selected

current file: latest MF, DF, or EF selected

Dedicated File (DF): file containing access conditions and, optionally, EFs or other DFs

directory: general term for MF and DF

Elementary File (EF): file containing access conditions and data and no other files

file: directory or an organized set of bytes or records in the SIM

file identifier: 2 bytes which address a file in the SIM

key generator: secure system entity authorized to generate Static Cipher Keys (SCKs) for Direct Mode Operation (DMO)

key holder: secure system entity authorized to distribute SCKs for DMO

key user: standard Direct Mode (DM) terminal which uses SCKs provided by an authorized key holder

ID-1 SIM: SIM having the format of an ID-1 card (see ISO 7816-1 [19])

input: signifies data input to the SIM functions (defined in clause 8):

Input from SIM input from the SIM internal memory;

Input from EF internal input from an EF on the SIM;

Input from ME data contained in a command APDU passed across the SIM-ME interface.

Master File (MF): unique mandatory DF representing the root

Mobile equipment (ME): part of the MS which interfaces to the SIM card

Mobile Station (MS): entirety of the equipment needed to communicate with the infrastructure (in trunked mode of operation) or direct with another MS (in direct mode of operation)

output: signifies data output from the SIM functions (defined in clause 8):

Output to SIM data shall be stored on the SIM in non-permanent memory for the duration of the TETRA session;

Output to EF internal updating of an EF on the SIM;

Output to ME data contained in a response APDU passed across the SIM-ME interface.

padding: one or more bits appended to a message in order to cause the message to contain the required number of bits or bytes

personalization: addition of subscriber and end user data to the appropriate EFs in the SIM during the administrative phase of a card's life cycle

pre-personalization: assignment of EF values at the manufacturing phase of a card's life cycle

plug-in SIM: second format of SIM (specified in clause 4)

record: string of bytes within an EF handled as a single entity (see clause 6)

record number: number which identifies a record within an EF

record pointer: pointer which addresses one record in an EF

Subscriber Identity Module (SIM) or SIM card: integrated circuit card containing network related subscriber information

T=0: half-duplex asynchronous character based transmission protocol. As defined in ISO/IEC 7816-3[21]

T=1: half-duplex asynchronous block based transmission protocol. The protocol may be initiated after ATR. As defined in ISO/IEC 7816-3[21]

TETRA application: set of security mechanisms, files, data and protocols required by TETRA

TETRA session: part of the card session dedicated to the TETRA operation

TETRA SIM: subscriber identity module used in a TETRA MS

usage phase: part of the card life, after the administrative phase, when the card is being used for operational purposes

5 V technology SIM: SIM operating at 5 V ±10 %

3 V technology SIM: SIM operating at 3 V ±10 % and 5 V ±10 %

3 V technology ME: ME operating the SIM - ME interface at 3 V ±10 % according to GSM 11.12 [15] and 5 V ±10 % according to GSM 11.11 [9]

3 V only ME: ME only operating the SIM - ME interface at 3 V ±10 % according to GSM 11.12 [15]

3.2 Symbols

For the purposes of the present document, the following symbols apply:

'0' to '9' and 'A' to 'F'	The sixteen hexadecimal digits
Vcc	Supply voltage
Vpp	Programming voltage

3.3 Abbreviations

For the purposes of the present document, the following abbreviations apply:

ADM	ADMinistrative (see definitions)
ADN	Abbreviated Dialling Number
ALW	ALWays
APDU	Application Protocol Data Unit
APN	Access Point Name
ASSI	Alias Short Subscriber Identity
ATR	Answer To Reset
AUTI	AUThorized Immediate (see definitions)
BCD	Binary Coded Decimal
CCK	Common Cipher Key
CCK-id	CCK identifier
CHV	Card Holder Verification (see definitions)
CLA	CLAss
CLK	CLocK
DCK	Derived Cipher Key
DCK1	Part 1 of the DCK
DCK2	Part 2 of the DCK
DF	Dedicated File
DGNA	Dynamic Group Number Assignment
DM	Direct Mode
DMO	Direct Mode Operation
DTMF	Dual Tone Multiple Frequency
EF	Elementary File
FDN	Fixed Dialling Number
FSSN	Fleet Specific Short Number
GCK	Group Cipher Key
GCKN	Group Cipher Key Number
GCK-VN	GCK Version Number
GGSN	Gateway GPRS Support Node
GPRS	General Packet Radio Service
GSSI	Group Short Subscriber Identity
GTSI	Group Tetra Subscriber Identity
I/O	Input/Output
IC	Integrated Circuit
ID	IDentifier
INS	
IP	Internet Protocol
ISSI	Individual Short Subscriber Identity
ITSI	Individual TETRA Subscriber Identity
K	individual subscriber authentication key
KE	Enhanced security Key
KSO	Sesson Key for Over The Air Re-keying
LA	Location Area
LND	Last Number Dialled
LSB	Least Significant Bit
MCC	Mobile Country Code
ME	Mobile Equipment
MF	Master File
MGCK	Modified Group Cipher Key

MMI	Man Machine Interface
MNC	Mobile Network Code
MS	Mobile Station
MSB	Most Significant Bit
MSISDN	Mobile Station ISDN Number
NET	NETwork
NEV	NEVer (see definitions)
NPI	Numbering Plan Identifier
OTAR	Over The Air Re-keying
PABX	Private Automatic Branch Exchange
PDU	Protocol Data Unit
PSTN	Public Switched Telephone Network
RAND1	RANDom challenge 1
RAND2	RANDom challenge 2
RES1	RESponse 1
RES2	RESponse 2
RFU	Reserved for Future Use
RS	Random Seed
RSO	Random Seed for OTAR
RST	ReSeT
SCCK	Sealed CCK
SCK	Static Cipher Key
SCKN	SCK number
SCK-VN	SCK version number
SDN	Service Dialling Number
SDS	Short Data Service
SDS-TL	Short Data Service - Transport Layer
SGCK	Sealed GCK
SIM	Subscriber Identity Module
SMS	Short Message Service (GSM)
SNA	Short Number Address
SSC	Supplementary Service Control string
SSCK	Sealed SCK
SSI	Short Subscriber Identity
SW1/SW2	Status Word 1/Status Word 2
SwMI	Switching and Management Infrastructure
TE	TETRA algorithm for enhanced security on SIM-ME interface
TON	Type Of Number
TP	Transfer layer Protocol
TSI	TETRA Subscriber Identity
UCS2	Universal Character Set 2
V-ASSI	Visitor Alias Short Subscriber Identity
XRES2	EXpected RESponse 2

4 SIM characteristics

Two physical types of SIM are specified. These are the "ID-1 SIM" (see ISO 7810 [16]) and the "Plug-in SIM" (see ENV 1375-1 [25]). The dimensions and mechanical characteristics of both types of physical SIM shall be in accordance with ISO/IEC 7816-1 [19] and ISO/IEC 7816-2 [20] unless otherwise specified. The following additional requirements shall be applied to ensure proper operation in the TETRA environment.

4.1 Format and layout

The identification number as defined in EF_{ICCID} (see clause 10.2.1) shall be present on the outside of the ID-1 card. The information on the outside of the plug-in SIM should include at least the individual account identifier and the check digit of the IC card Identification.

4.1.1 SIM

Format and layout of the ID-1 SIM shall be in accordance with ISO/IEC 7816-1 [19] and ISO/IEC 7816-2 [20].

The card shall have a polarization mark, which indicates how the user should insert the card into the Mobile Equipment (ME).

The ME shall accept embossed ID-1 cards. The embossing shall be in accordance with ISO 7811-1 [17] and ISO 7811-3 [18]. The contacts of the ID-1 SIM shall be located on the front (embossed face, see ISO 7810 [16]) of the card.

4.1.2 Plug-in SIM

The plug-in SIM has a width of 25 mm, a height of 15 mm, a thickness the same as an ID-1 SIM and a feature for orientation. See annex A, figure A.1 for details of the dimensions of the card and the dimensions and location of the contacts.

Annexes A.1 and A.2 of ISO 7816-1 [19] do not apply to the plug-in SIM.

Annex A of ISO 7816-2 [20] applies with the location of the reference points adapted to the smaller size. The three reference points P1, P2 and P3 measure 7,5 mm, 3,3 mm and 20,8 mm, respectively, from 0. The values in table A.1 of ISO 7816-2 [20] are replaced by the corresponding values of figure A.1.

4.2 Temperature range for card operation

The temperature range for full operational use shall be between -25°C and +70°C with occasional peaks of up to +85°C. "Occasional" means not more than 4 hours each time and not over 100 times during the life time of the card.

4.3 Contacts

4.3.1 Provision of contacts

ME: There need not be any contacting elements in positions C4 and C8. Contact C6 need not be provided.

SIM: Contacts C4 and C8 need not be provided by the SIM. Contact C6 shall not be bonded in the SIM.

4.3.2 Activation and deactivation

The ME shall connect, activate and deactivate the SIM in accordance with the operating procedures specified in ISO/IEC 7816-3 [21].

For any voltage level, monitored during the activation sequence, or during the deactivation sequence following soft power-down, the order of the contact activation/deactivation shall be respected.

NOTE 1: Soft power switching is when the radio is powered down normally. This is in contrast to abnormal power down, for instance if the battery is removed during operation, so that the voltage sequence can not be respected. Soft power switching is not defined in the TETRA specification.

NOTE 2: It is recommended that whenever possible the deactivation sequence defined in ISO/IEC 7816-3 [21] should be followed by the ME on all occasions when the ME is powered down.

If the SIM clock is already stopped and is not restarted, the ME is allowed to deactivate all the contacts in any order, provided that all signals reach low level before Vcc leaves high level. If the SIM clock is already stopped and is restarted before the deactivation sequence, then the deactivation sequence specified in ISO/IEC 7816-3 [21] clause 5.4 shall be followed.

4.3.3 Inactive contacts (contact conditions in the ME switched-off state)

The voltages on contacts C1, C2, C3, C6 and C7 of the ME shall be between 0 and ±0,4V referenced to ground (C5) when the ME is switched off with the power source connected to the ME. The measurement equipment shall have a resistance of 50 kΩ when measuring the voltage on C2, C3, C6 and C7. The resistance shall be 10 kΩ when measuring the voltage on C1.

4.3.4 Contact pressure

The contact pressure shall be large enough to ensure reliable and continuous contact (e.g. to overcome oxidization and to prevent interruption caused by vibration). The radius of any curvature of the contacting elements shall be greater than or equal to 0,8 mm over the contact area.

Under no circumstances may a contact force be greater than 0,5 N per contact.

4.4 Precedence (multiple SIM operation)

For Mobile Equipment, which accepts multiple SIMs, the user shall have the ability, via MMI, to select their preference for SIM Card Reader selection. The preferences shall initially have a default setting which the user can adjust. Once adjusted the preferences become the default setting.

If a non-preferred SIM Card Reader is in use and a SIM becomes available in a preferred SIM Card Reader, the user shall be offered the choice to select the preferred SIM.

On power up, the ME will attempt to use the user-preferred choice, but, if this is not available, the ME will use any other available SIM.

The use of a default setting will remove the necessity for the majority of users to access this MMI feature.

If a new SIM is inserted in the preferred reader during a call, the user shall be offered the ability to select the new SIM after the call is terminated.

If the active SIM is removed, any network transaction using that SIM shall be terminated immediately, as defined in GSM 11.11 [9] If another SIM is available, it shall be selected.

The ME shall indicate to the user which SIM is active.

4.5 Static protection

The ME manufacturer shall take adequate precautions (in addition to the protection diodes inherent in the SIM) to safeguard the ME, SIM and SIM/ME interface from static discharges at all times, and particularly during SIM insertion into the ME.

5 Electronic signals and transmission protocols

Electronic signals and transmission protocols shall be in accordance with ISO/IEC 7816-3 [21] unless specified otherwise. The following additional requirements shall be applied to ensure proper operation in the TETRA environment.

The choice of the transmission protocol(s), to be used to communicate between the SIM and the ME, shall at least include that specified and denoted by T = 0 in ISO/IEC 7816-3 [21].

As an option the T = 1 protocol specified in ISO/IEC 7816-3 [21] may also be supported.

5.1 Supply voltage Vcc (contact C1)

5 V and 3 V technology SIMs are defined in clause 3.1.

5.1.1 5 V technology SIM

The TETRA SIM shall operate on 5 V ± 10 % according to GSM 11.11 [9]. The electrical characteristics of Vcc and Icc under normal and transient operating conditions are defined in GSM 11.11 [9].

5.1.2 3 V technology SIM

The SIM shall operate on both 5 V ± 10 % according to GSM 11.11 [9], and on 3 V ± 10 % according to GSM 11.12 [15]. If the ME supplies 5 V to the SIM, both the ME and the SIM shall operate according to GSM 11.11 [9]. The logical operation of the 3 V technology SIM shall be as defined in GSM 11.11 [9].

Clock stop mode shall be supported by the SIM. The SIM shall indicate "Clock Stop Allowed" in the file characteristics of the status information as specified in GSM 11.11 [9].

5.1.3 3 V technology SIM identification

The 3 V technology SIM shall contain an identification. The identification is coded on bit 5 in byte 14 of the status information (see clause 9.2.1) as follows:

"0" : 5 V only SIM;

"1" : 3 V technology SIM.

In the case that the ME offers full compatibility by being able to operate the SIM interface at both 3 V and 5 V, then bit 5 in byte 14 of the status information, when set to "1", indicates that the SIM may be operated at 3 V.

The procedure for deriving the identification bit shall be performed by the ME immediately after the Answer To Reset (ATR) and before issuing any other command. The procedure consists of the two commands "SELECT TETRA" and "STATUS/GET RESPONSE".

5.1.4 3 V technology ME

The 3 V technology ME shall initially activate the SIM with 5 V according to GSM 11.11 [9]. If the SIM indicates 3 V operation as defined in clause 5.1.3, the ME may switch to 3 V operation as defined in clause 5.1.7. If switching is performed it shall take place before issuing any further commands.

5.1.5 3 V Only ME

The 3 V only ME activates the SIM at 3 V.

If the ME is able to detect a 5 V only SIM according to the procedure in clause 5.1.3, or if the procedure cannot be completed, the ME shall deactivate and reject the SIM immediately (maximum of 5 s) without issuing any further command. This rejection ensures that a SIM, which appears to operate successfully during these early procedures, is not allowed to continue further into the TETRA session where it may subsequently give unreliable operation at 3 V.

5.1.6 Activation and deactivation of 3 V technology SIM

The ME shall connect, activate and deactivate the 3 V SIM in accordance with the operating procedures specified in GSM 11.11 [9] taking into account the electrical characteristics specified in GSM 11.12 [15]. In particular, Vcc is powered when it has a value between 2,7 V and 3,3 V.

5.1.7 Supply voltage switching

MEs supporting both 3 V and 5 V operation may switch between the two supply voltages. Switching shall always be performed by deactivating the SIM and activating it at the new supply voltage. Activation and deactivation of the SIM with 5 V shall be according to GSM 11.11 [9], whereas activation and deactivation of the SIM with 3 V shall be according to GSM 11.12 [15].

5.1.8 Cross compatibility

Cross compatibility means that the ME supports 3 V and 5 V operation. This is, however, optional for the ME. In case of the 3 V technology ME, full cross compatibility is provided, whereas, a 3 V only ME requires a 3 V technology SIM for operation. However, the 3 V technology SIM (see definitions) ensures full cross compatibility.

5.1.9 Technology outlook

Due to technology development it is possible in the future, when sub-micron technology is introduced, that ICs used in MEs may not withstand the 5 V supply voltage. This may, in particular, be the case for ICs operating in the power supply range of 1,5 V to 3,6 V. It may therefore be necessary in the future to specify a low voltage only SIM interface.

NOTE: When a low voltage only SIM is inserted into a ME, which is supplying 5 V, the SIM may be destroyed. In some cases this could cause permanent damage to the ME. Precautions should be taken by the IC manufacturers to prevent the low voltage ICs from being damaged at 5 V.

5.2 Reset (RST) (contact C2)

For 5 V operation the ME shall operate the SIM within the limits defined in GSM 11.11 [9].

For 3 V operation the ME shall operate the SIM within the limits defined in GSM 11.12 [15].

5.3 Programming voltage Vpp (contact C6)

The SIM need not provide contact C6. If the SIM provides contact C6, then contact C6 shall not be connected.

5.4 Clock CLK (contact C3)

The clock shall be supplied by the ME. No "internal clock" SIMs shall be used.

When supplied with the 5 V ± 10 % supply voltage, as specified in GSM 11.11 [9], the SIM shall support 1 MHz to 5 MHz clock frequency operation.

When supplied with the 3 V ± 10 % supply voltage, as specified in GSM 11.12 [15], the SIM shall be operated with a clock frequency of 1 MHz to 4 MHz.

The required electrical characteristics of the ME clock for 5 V operation are defined in GSM 11.11 [9]. For 3 V operation defined in GSM 11.11 [9] is replaced by the definition in GSM 11.12 [15].

5.5 Input/Output (I/O) (contact C7)

For 5 V operation the electrical characteristics of the I/O (contact C7) are defined in GSM 11.11 [9].

For 3 V operation the electrical characteristics of the I/O (contact C7) are defined in GSM 11.12 [15].

5.6 States

There are two states for the SIM while the power supply is on:

- the SIM is in operating state when it executes a command. This state also includes transmission from and to the ME;
- the SIM is in idle state at any other time. It shall retain all pertinent data during this state.

The SIM may support a clock stop mode. The clock shall only be switched off subject to the conditions specified in the directory characteristics (see clause 8.17.2).

Clock stop mode:

A ME shall wait at least 1 860 clock cycles after having received the last character of the response, including guard time (744 clock cycles), before it switches off the clock (if it is allowed to do so). It shall wait at least 744 clock cycles before it sends the first command after having started the clock.

5.7 Baud rate

The baud rate for all communications shall be as defined in GSM 11.11 [9].

5.8 Answer To Reset (ATR)

The ATR is information presented by the SIM to the ME at the beginning of the card session and gives operational requirements.

The table explaining the structure and content of the ATR characters (as specified in ISO/IEC 7816-3 [21]) and the requirements for their use in TETRA, follows that for GSM as defined in GSM 11.11 [9].

The protocol type selection procedures and speed selection procedures defined in GSM 11.11 [9] shall apply to the TETRA SIM.

5.9 Bit/character duration and sampling time

The bit/character duration and sampling time specified in ISO/IEC 7816-3 [21], are valid for all communications.

5.10 Error handling

If an ATR is corrupted or not received by the ME, error handling according to GSM 11.11 [9] shall apply.

6 Logical model

This clause describes the logical structure for a SIM, the code associated with it, and the structure of files used.

6.1 General description

Figure 1 shows the general structural relationships, which may exist, between files. The files are organized in a hierarchical structure and are of one of three types as defined below. These files may be either administrative or application specific. The operating system handles the access to the data stored in different files.

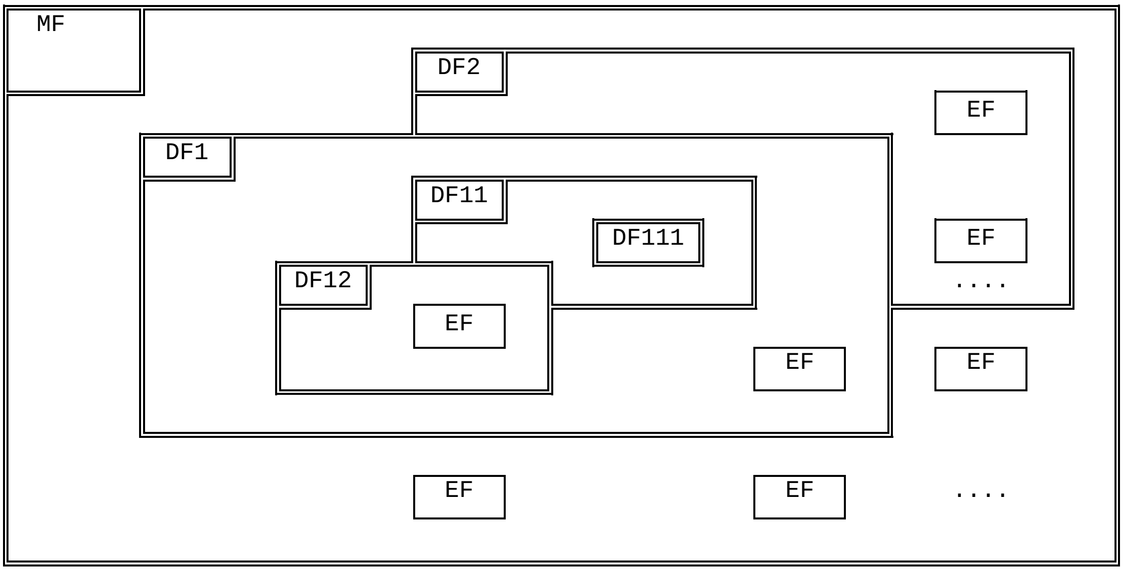


Figure 1: Organization of memory

Files are composed of a header, which is internally managed by the SIM, and optionally a body part. The information of the header is related to the structure and attributes of the file and may be obtained by using the commands "GET RESPONSE" or "STATUS". This information is fixed during the administrative phase. The body part contains the data of the file.

6.2 File identifier

A file IDentifier (ID) is used to address or identify each specific file. The file ID consists of two bytes and shall be coded in hexadecimal notation. They are specified in clause 10.

The first byte identifies the type of file, and for TETRA is:

- '3F': Master File;
- '7F': 1st level Dedicated File;
- '5F': 2nd level Dedicated File;
- '2F': Elementary File under the Master File;
- '6F': Elementary File under a 1st level Dedicated File;
- '4F': Elementary File under 2nd level Dedicated File.

File IDs shall be subject to the following conditions:

- the file ID shall be assigned at the time of creation of the file concerned;
- no two files under the same parent shall have the same ID;
- a child and any parent, either immediate or remote in the hierarchy, e.g. grandparent, shall never have the same file ID.

In this way each file is uniquely identified.

6.3 Dedicated Files (DF)

A DF is a functional grouping of files consisting of itself and all those files which contain this DF in their parental hierarchy (that is to say it consists of the DF and its complete "subtree"). A DF "consists" only of a header part.

Two 1st level DFs are defined in the present document:

- DF_{TETRA} which contains the application for TETRA;
- $DF_{TELECOM}$ which contains telecom service features.

Both files are immediate children of the MF and may coexist on a multi-application card.

2nd level DFs are defined in the present document under DF_{TETRA}.

All 2nd level DFs are immediate children of the DF_{TETRA}.

6.4 Elementary Files (EF)

An EF is composed of a header and a body part. The following clauses give the four structures of an EF are used by TETRA.

6.4.1 Transparent EF

An EF with a transparent structure consists of a sequence of bytes. When reading or updating, the sequence of bytes to be acted upon, is referenced by a relative address (offset), which indicates the start position (in bytes), and the number of bytes to be read or updated. The first byte of a transparent EF has the relative address '00 00'. The total data length of the body of the EF is indicated in the header of the EF.

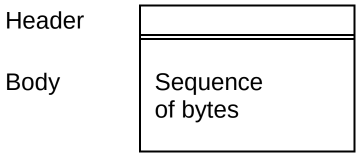


Figure 2: Structure of a transparent EF

6.4.2 Linear fixed EF

An EF with linear fixed structure consists of a sequence of records all having the same (fixed) length. The first record is record number 1. The length of a record, as well as this value multiplied by the number of records, is indicated in the header of the EF.

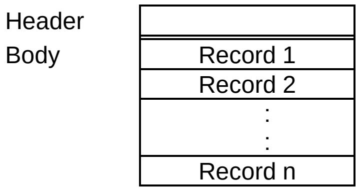


Figure 3: Structure of a linear fixed file

There are several methods to access records within an EF of this type:

- absolutely, using the record number;
- when the record pointer is not set it shall be possible to perform an action on the first or the last record;
- when the record pointer is set it shall be possible to perform an action on this record, the next record (unless the record pointer is set to the last record) or the previous record (unless the record pointer is set to the first record);
- by identifying a record using pattern seek starting:
  - forwards from the beginning of the file;
  - forwards from the record following the one at which the record pointer is set (unless the record pointer is set to the last record);
  - backwards from the end of the file;
  - backwards from the record preceding the one at which the record pointer is set (unless the record pointer is set to the first record).

If an action following selection of a record is aborted, then the record pointer shall remain set at the record at which it was set prior to the action.

NOTE: It is not possible, at present, to have more than 255 records in a file of this type, and each record cannot be greater than 255 bytes.

6.4.3 Key EF

A key EF consists of a sequence of records all having the same (fixed) length. The first record is record number 1. The length of a record, as well as this value multiplied by the number of records, are indicated in the header of the EF.

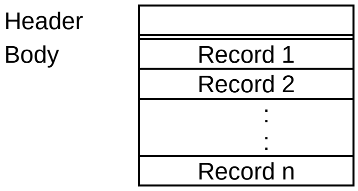


Figure 4: Structure of a key file

Records in an EF of this type are accessed using the absolute record number.

NOTE: It is not possible to have more than 255 records in a file of this type, and each record cannot be greater than 255 bytes.

6.4.4 Cyclic EF

Cyclic files are used for storing records in chronological order. When all records have been used for storage, then the next storage of data shall overwrite the oldest information.

An EF with a cyclic structure consists of a fixed number of records with the same (fixed) length. In this file structure there is a link between the last record (n) and the first record. When the record pointer is set to the last record n, then the next record is record 1. Similarly, when the record pointer is set to record 1, then the previous record is record n. The last updated record containing the newest data is record number 1, and the oldest data is held in record number n.

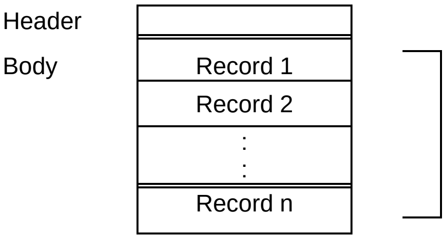


Figure 5: Structure of a cyclic file

For update operations only PREVIOUS record shall be used. For reading operations, the methods of addressing are "NEXT", "PREVIOUS", "CURRENT" and "RECORD NUMBER".

After selection of a cyclic file (for either operation), the record pointer shall address the record updated or increased last. If an action following selection of a record is aborted, then the record pointer shall remain set at the record at which it was set prior to the action.

NOTE: It is not possible, at present, to have more than 255 records in a file of this type, and each record cannot be greater than 255 bytes.

6.5 Methods for selecting a file

After the ATR, the MF is implicitly selected and becomes the "Current Directory". Each file may then be selected by using the SELECT function in accordance with the following rules:

- selecting a DF or the MF sets the "Current Directory";

  after such a selection there is no current EF;

- selecting an EF sets the current EF, and the "Current Directory" remains the DF or MF which is the parent of this EF;

  the current EF is always a child of the "Current Directory".

Any application specific command shall only be operable if it is specific to the "Current Directory".

The following files may be selected from the last selected file:

- any file which is an immediate child of the "Current Directory";
- any DF which is an immediate child of the parent of the current DF;
- the parent of the "Current Directory";
- the current DF;
- the MF.

This means in particular that a DF shall be selected prior to the selection of any of its EFs. All selections are made using the file ID.

The following figure gives the logical structure for the TETRA application. TETRA defines only two levels of DFs under the MF.

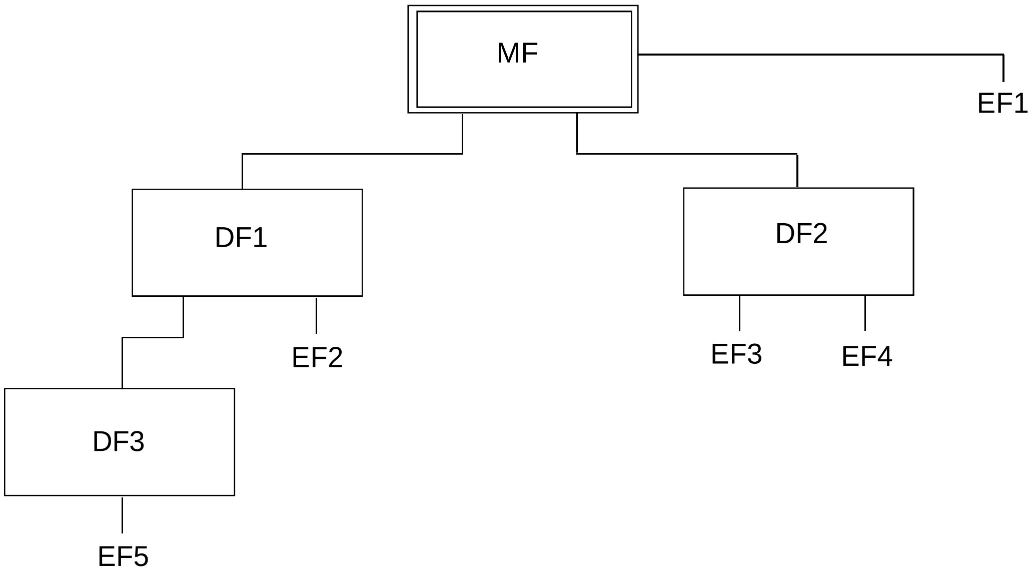


Figure 6: Logical structure

Table 1 gives the valid selections for TETRA for the logical structure in figure 6. Reselection of the last selected file is also allowed but not shown.

Table 1: File selection

Last selected file	Valid Selections
MF	DF1, DF2, EF1
DF1	MF, DF2, DF3, EF2
DF2	MF, DF1, EF3, EF4
DF3	MF, DF1, EF5
EF1	MF, DF1, DF2
EF2	MF, DF1, DF2, DF3
EF3	MF, DF1, DF2, EF4
EF5	MF, DF1, DF3

6.6 Reservation of file IDs

In addition to the identifiers used for the files specified in the present document, the following file IDs are reserved for use by TETRA.

DF:

- administrative use:

  '7F 8X';

- operational use:

  '7F 10' ($DF_{TELECOM}$), '7F 90' (DF_{TETRA}),

EFs:

- administrative use:

  '6F XX' in the DFs '7F 8X';

  '6F CX' in the DFs '7F90';'7F10'

  '2F XX ', in the MF '3F 00';

- operational use:

  '6F XX in the DFs '7F 90'and '7F10';

  '2F 1X' in the MF '3F 00'.

In all the above cases X ranges from '0' to 'F', unless otherwise stated.

The value 'FF FF' shall not be used.

NOTE: When choosing file IDs, care should be taken to avoid conflicts with IDs already used in other standards concerning IC cards for telecommunications use.

7 Security features

The security aspects of TETRA are described in EN 300 392-7 [3] and ETS 300 396-6 [7]. This clause gives information related to security features supported by the SIM to enable the following:

- authentication of the subscriber identity to the network;
- data confidentiality over the air interface;
- confidentiality of air interface keys when passed over the SIM-ME interface;
- file access conditions.

The security of an MS is defined by security class (EN 300 392-7 [3]). The table below indicates for which class the SIM has to provide security functions and key storage.

Class	Authentication	Key store	OTAR SCK	OTAR GCK	OTAR CCK
1	O	n/a	n/a	n/a	n/a
2	O	SCK	O	n/a	n/a
3	M	DCK, CCK, GCK, MGCK	O	O	M
Note 1: Where authentication is provided the SIM shall also store K (not in an accessible EF). Note 2: M = Mandatory, O = Optional and n/a = not applicable					

7.1 Authentication and cipher key generation procedure

This clause describes the authentication mechanism and cipher key generation which are invoked by the network and the SIM.

The names and parameters of the authentication algorithms supported by the SIM are defined in EN 300 392-7 [3]. These are:

- algorithms TA11/TA12 to authenticate the SIM to the SwMI;
- algorithms TA21/TA22 to authenticate the SwMI to the SIM.

The cipher key generation algorithm supported by the SIM is defined in EN 300 392-7 [3] and is required only for a SIM-ME pair supporting Class 3 security. This is:

- algorithm TB4 to generate the Derived Cipher Key (DCK).

These algorithms may exist either discretely or combined within the SIM.

7.2 Support of Over The Air Re-keying (OTAR) distribution of cipher keys

The names and parameters of the OTAR algorithms supported by the SIM are defined in EN 300 392-7 [3] and ETS 300 396-6 [7]. These are:

- algorithm TA32 to obtain the Common Cipher Key (CCK) from the Sealed CCK (SCCK);
- algorithm TA41/TA82 to obtain the Group Cipher Key (GCK) from the Sealed Group Cipher Key (SGCK);
- algorithm TA41/TA52 to obtain the Static Cipher Key (SCK) from the Sealed SCK (SSCK);
- algorithm TA71 to obtain the Modified Group Cipher Key (MGCK) from the GCK.

These algorithms may exist either discretely or combined within the SIM.

7.3 Support of SIM-ME enhanced security

Enhanced security for DCK, CCK, SCK and MGCK on the SIM-ME interface in SIM-ME pairs supporting security Class 2 and 3 is supported by use of the TETRA algorithm for enhanced security on SIM-ME interface (TE) algorithm. When enhanced SIM-ME security is required (SIM Service 20 set):

- algorithm immediately following TB4 algorithm
  - CCK, SCK and MGCK are sealed by the TE algorithm as part of the "Read Key" command (see clause 8.6).

7.4 File access conditions

Every file has its own specific access condition for each command. The relevant access condition of the last selected file shall be fulfilled before the requested action can take place. Access conditions are defined in clause 3.1.

For each file:

- the access conditions for the commands READ and SEEK are identical;
- the access conditions for the commands SELECT and STATUS are ALWays.

The access condition levels are defined in table 2.

Table 2: Access condition level coding

Level	Access Condition
0	ALWays
1	CHV1
2	CHV2
3	AUTI
4 to 14	ADM
15	NEVer

NEV: The action cannot be performed over the SIM/ME interface. The SIM may perform the action internally.

AUTI: The mobile can perform the action over the SIM/ME but only if it is the next action over the SIM/ME interface following a successful authentication of the SwMI by the terminal.

So far as the SIM/ME interface is concerned the status AUTI shall be granted immediately following a successful authentication (as indicated by the return code on running the TA21/22 ALGORITHM). The AUTI status shall be withdrawn when there is subsequent activity over the SIM/ME interface. The only command not changing the current status of the AUTI access condition is the SELECT command. During an authentication message exchange, the ME may need to manage presence check procedures (for instance sending a SIM presence check immediately prior to initiating the authentication) so that these messages do not invalidate the AUTI status.

The AUTI access condition applies to the following Elementary Files and for the identified actions.

EF	Action
ITSI	Invalidate
ITSIDIS	Update
GSSID	Update, Invalidate

CHV1: The action shall only be possible if one of the following three conditions is fulfilled:

- a correct CHV1 value has already been presented to the SIM during the current card session;
- the CHV1 enabled/disabled indicator is set to "disabled";
- UNBLOCK CHV1 has been successfully performed during the current card session.

CHV2: The action shall only be possible if one of the following two conditions is fulfilled:

- a correct CHV2 value has already been presented to the SIM during the current card session;
- UNBLOCK CHV2 has been successfully performed during the current card session.

ADM: Allocation of these levels and the respective requirements for their fulfilment are the responsibility of the appropriate administrative authority.

The definition of access condition ADM does not preclude the administrative authority from using ALW, CHV1, CHV2 and NEV if required.

ALW: The action is always possible.

Access condition levels are not hierarchical. For instance, correct presentation of CHV2 does not allow actions to be performed, which require presentation of CHV1. A condition level, which has been satisfied, remains valid until the end of the TETRA session as long as the corresponding secret code remains unblocked.

The ME shall determine whether CHV2 is available by using the response to the STATUS command. If CHV2 is "not initialized" then CHV2 commands, e.g. VERIFY CHV2, shall not be executable.

NOTE: The personalization phase of the card is normally agreed between the manufacturer and the network operator. Security during this phase is outside the scope of the present document and needs to be carefully controlled.

7.5 Storage of DCK

After successful authentication DCK shall be stored on the SIM for further use to unseal cipher keys but only for the duration of the TETRA session.

8 Description of the functions

This clause gives a functional description of the commands and their respective responses. Associated status conditions, error codes and their corresponding coding are specified in clause 9.2.

Cards complying with the present document shall support all functions described in this clause. In addition the command GET RESPONSE (specified in clause 9.2.24) which is needed for the protocol T = 0 shall be supported.

The following table lists the file types and structures together with the functions which may act on them during a TETRA session. These are indicated by an asterisk (*).

Table 3: Functions which operate on files in a TETRA session

	File					
Function	**MF**	**DF**	**EF transparent**	**EF linear fixed**	**EF cyclic**	**EF key**
SELECT	*	*	*	*	*	
STATUS	*	*	*	*	*	
READ BINARY			*			
UPDATE BINARY			*			
READ RECORD				*	*	
UPDATE RECORD				*	*	
SEEK				*		
INVALIDATE			*	*	*	
REHABILITATE			*	*	*	
READ KEY						*

The commands and responses are defined in terms of where they obtain their inputs and where they place their outputs. The definitions of inputs and outputs to or from SIM, EF or ME are given in clause 3.

8.1 SELECT

This function selects a file according to the methods described in clause 6.5. After a successful selection the record pointer in a linear fixed file is undefined. The record pointer in a cyclic file shall address the last record which has been updated.

Input from ME: file ID.

Output to ME:

- if the selected file is the MF or a DF:

  file ID, total memory space available, CHV enabled/disabled indicator, CHV status;

- if the selected file is an EF:

  file ID, file size, access conditions, invalidated/not invalidated indicator, structure of EF and length of the records in case of linear fixed structure or cyclic structure.

8.2 STATUS

This function returns information concerning the current directory. A current EF is not affected by the STATUS function.

Input from ME: none.

Output to ME: file ID, total memory space available, CHV enabled/disabled indicator, CHV status.

8.3 READ BINARY

This function reads a string of bytes from the current transparent EF. This function shall only be performed if the READ access condition for this EF is satisfied.

Input from ME: relative address and the length of the string.

Output to ME: string of bytes.

8.4 UPDATE BINARY

This function updates the current transparent EF with a string of bytes. This function shall only be performed if the UPDATE access condition for this EF is satisfied. An update can be considered as a replacement of the string already present in the EF by the string given in the update command.

Input from ME: relative address and the length of the string; string of bytes.

Output to ME: none.

8.5 READ RECORD

This function reads one complete record in the current linear fixed or cyclic EF. The record to be read is described by the modes below. This function shall only be performed if the READ access condition for this EF is satisfied. The record pointer shall not be changed by an unsuccessful READ RECORD function.

Four modes are defined.

1) **CURRENT:**

   the current record is read. The record pointer is not affected.

2) **ABSOLUTE:**

   the record given by the record number is read. The record pointer is not affected.

3) **NEXT:**

   the record pointer is incremented before the READ RECORD function is performed and the pointed record is read. If the record pointer has not been previously set within the selected EF, then READ RECORD (next) shall read the first record and set the record pointer to this record.

   If the record pointer addresses the last record in a linear fixed EF, READ RECORD (next) shall not cause the record pointer to be changed, and no data shall be read.

   If the record pointer addresses the last record in a cyclic EF, READ RECORD (next) shall set the record pointer to the first record in this EF and this record shall be read.

4) **PREVIOUS:**

the record pointer is decremented before the READ RECORD function is performed and the pointed record is read. If the record pointer has not been previously set within the selected EF, then READ RECORD (previous) shall read the last record and set the record pointer to this record.

If the record pointer addresses the first record in a linear fixed EF, READ RECORD (previous) shall not cause the record pointer to be changed, and no data shall be read.

If the record pointer addresses the first record in a cyclic EF, READ RECORD (previous) shall set the record pointer to the last record in this EF and this record shall be read.

Input from ME: mode, record number (absolute mode only) and the length of the record.

Output to ME: the record.

8.6 READ KEY

This function reads one complete record in the current key EF. The record to be read is described by the mode below.

ABSOLUTE:

the record given by the record number is read.

If SIM Service 20 is set (Enhanced SIM-ME security) the enhanced security algorithm TE shall be automatically run by the SIM to seal the OTAR keys (MGCK, CCK or SCK) with Enhanced Security Key (for protection of OTAR information on SIM-ME interface) (KE) before sending them to the ME.

Input from ME: record number and the length of the record.

Input from SIM: optionally KE (if SIM Service 20 is set).

Output to ME: the record (sealed by KE if service 20 is set).

8.7 UPDATE RECORD

This function updates one complete record in the current linear fixed or cyclic EF. This function shall only be performed if the UPDATE access condition for this EF is satisfied. The UPDATE can be considered as a replacement of the relevant record data of the EF by the record data given in the command. The record pointer shall not be changed by an unsuccessful UPDATE RECORD function.

The record to be updated is described by the modes below. Four modes are defined of which only PREVIOUS is allowed for cyclic files:

CURRENT:

the current record is updated. The record pointer is not affected.

ABSOLUTE:

the record given by the record number is updated. The record pointer is not affected.

NEXT:

the record pointer is incremented before the UPDATE RECORD function is performed and the pointed record is updated. If the record pointer has not been previously set within the selected EF, then UPDATE RECORD (next) shall set the record pointer to the first record in this EF and this record shall be updated. If the record pointer addresses the last record in a linear fixed EF, UPDATE RECORD (next) shall not cause the record pointer to be changed, and no record shall be updated.

PREVIOUS:

for a linear fixed EF the record pointer is decremented before the UPDATE RECORD function is performed and the pointed record is updated. If the record pointer has not been previously set within the selected EF, then UPDATE RECORD (previous) shall set the record pointer to the last record in this EF and this record shall be updated. If the record pointer addresses the first record in a linear fixed EF, UPDATE RECORD (previous) shall not cause the record pointer to be changed, and no record shall be updated.

For a cyclic EF the record containing the oldest data is updated, the record pointer is set to this record and this record becomes record number 1.

Input from ME:

- mode, record number (absolute mode only) and the length of the record;
- the data used for updating the record.

Output to ME: none.

8.8 SEEK

This function searches through the current linear fixed EF to find a record starting with the given pattern. This function shall only be performed if the READ access condition for this EF is satisfied. Two types of SEEK are defined:

Type 1:

the record pointer is set to the record containing the pattern, no output is available.

Type 2:

the record pointer is set to the record containing the pattern, the output is the record number.

The SIM shall be able to accept any pattern length from 1 to 16 bytes inclusive. The length of the pattern shall not exceed the record length.

Four modes are defined:

- from the beginning forwards;
- from the end backwards;
- from the next location forwards;
- from the previous location backwards.

If the record pointer has not been previously set (its status is undefined) within the selected linear fixed EF, then the search begins:

- with the first record in the case of SEEK from the next location forwards; or
- with the last record in the case of SEEK from the previous location backwards.

After a successful SEEK, the record pointer is set to the record in which the pattern was found. The record pointer shall not be changed by an unsuccessful SEEK function.

Input from ME:

- type and mode;
- pattern;
- length of the pattern.

Output to ME:

- type 1: none;
- type 2: status/record number.

8.9 VERIFY CHV

This function verifies the CHV presented by the ME by comparing it with the relevant one stored in the SIM. The verification process is subject to the following conditions being fulfilled:

CHV is not disabled;

CHV is not blocked.

If the access condition for a function to be performed on the last selected file is CHV1 or CHV2, then a successful verification of the relevant CHV is required prior to the use of the function on this file unless the CHV is disabled.

If the CHV presented is correct, the number of remaining CHV attempts for that CHV shall be reset to its initial value 3.

If the CHV presented is false, the number of remaining CHV attempts for that CHV shall be decremented. After 3 consecutive false CHV presentations, not necessarily in the same card session, the respective CHV shall be blocked and the access condition can never be fulfilled until the UNBLOCK CHV function has been successfully performed on the respective CHV.

Input from ME: indication CHV1/CHV2, CHV.

Output to ME: none.

8.10 CHANGE CHV

This function assigns a new value to the relevant CHV subject to the following conditions being fulfilled:

CHV is not disabled;

CHV is not blocked.

The old and new CHV shall be presented.

If the old CHV presented is correct, the number of remaining CHV attempts for that CHV shall be reset to its initial value 3 and the new value for the CHV becomes valid.

If the old CHV presented is false, the number of remaining CHV attempts for that CHV shall be decremented and the value of the CHV is unchanged. After 3 consecutive false CHV presentations, not necessarily in the same card session, the respective CHV shall be blocked and the access condition can never be fulfilled until the UNBLOCK CHV function has been performed successfully on the respective CHV.

Input from ME: indication CHV1/CHV2, old CHV, new CHV.

Output to ME: none.

8.11 DISABLE CHV

This function may only be applied to CHV1. The successful execution of this function has the effect that files protected by CHV1 are now accessible as if they were marked "ALWAYS". The function DISABLE CHV shall not be executed by the SIM when CHV1 is already disabled or blocked.

If the CHV1 presented is correct, the number of remaining CHV1 attempts shall be reset to its initial value 3 and CHV1 shall be disabled.

If the CHV1 presented is false, the number of remaining CHV1 attempts shall be decremented and CHV1 remains enabled. After 3 consecutive false CHV1 presentations, not necessarily in the same card session, CHV1 shall be blocked and the access condition can never be fulfilled until the UNBLOCK CHV function has been successfully performed on CHV1.

Input from ME: CHV1.

Output to ME: none.

8.12 ENABLE CHV

This function may only be applied to CHV1. It is the reverse function of DISABLE CHV. The function ENABLE CHV shall not be executed by the SIM when CHV1 is already enabled or blocked.

If the CHV1 presented is correct, the number of remaining CHV1 attempts shall be reset to its initial value 3 and CHV1 shall be enabled.

If the CHV1 presented is false, the number of remaining CHV1 attempts shall be decremented and CHV1 remains disabled. After 3 consecutive false CHV1 presentations, not necessarily in the same card session, CHV1 shall be blocked and the access condition can never be fulfilled until the UNBLOCK CHV function has been successfully performed on CHV1.

Input from ME: CHV1.

Output to ME: none.

8.13 UNBLOCK CHV

This function unblocks a CHV which has been blocked by 3 consecutive wrong CHV presentations. This function may be performed whether or not the relevant CHV is blocked.

If the UNBLOCK CHV presented is correct, the value of the CHV, presented together with the UNBLOCK CHV, is assigned to that CHV, the number of remaining UNBLOCK CHV attempts for that UNBLOCK CHV is reset to its initial value 10 and the number of remaining CHV attempts for that CHV is reset to its initial value 3. After a successful unblocking attempt the CHV is enabled and the relevant access condition level is satisfied.

If the presented UNBLOCK CHV is false, the number of remaining UNBLOCK CHV attempts for that UNBLOCK CHV shall be decremented. After 10 consecutive false UNBLOCK CHV presentations, not necessarily in the same card session, the respective UNBLOCK CHV shall be blocked. A false UNBLOCK CHV shall have no effect on the status of the respective CHV itself.

Input from ME: indication CHV1/CHV2, the UNBLOCK CHV and the new CHV.

Output to ME: none.

8.14 INVALIDATE

This function invalidates the current EF. After an INVALIDATE function the respective flag in the file status shall be changed accordingly. This function shall only be performed if the INVALIDATE access condition for the current EF is satisfied.

An invalidated file shall no longer be available within the application for any function except for the SELECT and the REHABILITATE functions unless the file status of the EF indicates that READ and UPDATE may also be performed.

Input from ME:	none.
Output to ME:	none.

8.15 REHABILITATE

This function rehabilitates the invalidated current EF. After a REHABILITATE function the respective flag in the file status shall be changed accordingly. This function shall only be performed if the REHABILITATE access condition for the current EF is satisfied.

Input from ME:	none.
Output to ME:	none.

8.16 TETRA authentication algorithms

These functions support authentication of the Individual TETRA Subscriber Identity (ITSI) and of the Switching and Management Infrastructure (SwMI).

The algorithms shall not be executable unless DF_{TETRA} has been selected as the Current Directory and a successful CHV1 verification procedure has been performed.

Security procedures internal to the SIM shall ensure that the authentication algorithms can only be run in the order specified in EN 300 392-7 [3] and ETS 300 396-6 [7].

8.16.1 GET RANDOM

This function produces a random number for use in the authentication algorithms.

Input from ME:	none.
Output to ME:	RANDom challenge 2 (RAND2).
Output to SIM:	RAND2.

The result RAND2 shall be stored internally on the SIM and also output to the ME for onward transfer to the SwMI. RAND2 shall be used as the input random number for the SIM initiated authentication procedure TA21/TA22.

8.16.2 TA11/TA12 ALGORITHM

This function, initiated by the SwMI, is used for authenticating the SIM to the TETRA network (SwMI).

Input from ME:	RANDom challenge 1 (RAND1), Random Seed (RS).
Input from SIM:	K.
Output to SIM:	DCK1.
Output to ME:	Response 1 (RES1).

RES1 shall be obtained from the SIM by use of the GET RESPONSE command.

8.16.3 TA21/TA22 ALGORITHM

This function, initiated by the SIM, is used for authenticating the TETRA network (SwMI) to the SIM.

Input from ME:	Response 2 (RES2), RS.
Input from SIM:	K, RAND2.
Output to SIM:	DCK2.
Output to ME:	XRES2

XRES2 shall be obtained from the SIM by use of the GET RESPONSE command.

NOTE: The ME is informed about the success of the operation via the status condition [R2] returned by the SIM (see also clause 9.4.4).

8.16.4 TB4/TE ALGORITHM

This function is used to obtain the DCK from its two parts DCK1 and DCK2 by use of the specified algorithm TB4. If SIM Service 20 is set (enhanced SIM-ME security) the enhanced security algorithm TE is automatically run by the SIM to seal DCK with KE before sending it to the ME.

Input from SIM:	DCK1, DCK2, optionally KE (if SIM Service 20 is set).
Output to SIM:	DCK.
Output to ME:	DCK (sealed by KE if service 20 is set).

In the case of mutual authentication (SIM< = >SwMI) the inputs DCK1 and DCK2 shall be obtained internally from the TA11/TA12 and TA21/TA22 algorithms respectively. In the case of unilateral authentication, either DCK1 or DCK2 shall be set to zero; for SIM authentication DCK2 = 0; for SwMI authentication DCK1 = 0.

8.17 OTAR algorithms

These algorithms support the distribution of sealed cipher keys over the radio air interface using the OTAR procedures defined in EN 300 392-7 [3] and ETS 300 396-6 [7].

The algorithms shall not be executable unless DF_{TETRA} has been selected as the Current Directory and a successful CHV1 verification procedure has been performed.

Security procedures internal to the SIM shall ensure that the OTAR algorithms can only be run in the order specified in EN 300 392-7 [3] and ETS 300 396-6 [7].

8.17.1 TA32 ALGORITHM

This function is used to obtain the CCK from the SCCK by use of the specified algorithm TA32. The SCCK can be delivered to the ME in sealed format by an OTAR procedure. The SCCK shall be unsealed on the SIM and the CCK stored on the SIM for subsequent use in the ME.

Input from ME:	SCCK, CCK-id.
Input from SIM:	DCK.
Output to EF:	CCK, CCK-id.
Output to ME:	None

NOTE: The ME is informed about the success of the operation via the status condition (manipulation flag) returned by the SIM (see also clause 9.4.4).

8.17.2 TA41/TA82 ALGORITHM

This function shall be used to compute GCK and GCKN from SGCK, GCK Version Number (GCK-VN) and KSO.

Input from ME: SGCK, GCK-VN, Random Seed for OTAR (RSO)

Input from SIM: K.

Output to EF: GCK (to EF_{GCK}), GCKN

Output to ME: None.

NOTE 1: GCKs are not accessible over the SIM-ME interface.

Following the download of a new GCK, algorithm TA71 (see clause 8.17.4) is run to update the associated MGCK.

NOTE 2: The ME is informed about the success of the operation via the status condition (manipulation flag) returned by the SIM (see also clause 9.4.4).

8.17.3 TA41/TA52 ALGORITHM

This function is used to obtain the SCK from the SSCK which may be distributed by OTAR. The SSCKs shall be unsealed on the SIM and the SCK stored on the SIM for subsequent use in the ME.

Input from ME: SSCK, SCK-VN, Random Seed for OTAR (RSO).

Input from SIM: K.

Output to EF: SCK, SCKN.

Output to ME: None.

NOTE: The ME is informed about the success of the operation via the status condition (manipulation flag) returned by the SIM (see also clause 9.4.4).

Algorithm TA52 shall output SCKN which shall be used as an index to the record in $EF_{SCK.}$ The record number shall be updated only if the Manipulation flag is TRUE.

8.17.4 TA71 ALGORITHM

This function shall be used to obtain the MGCK from the GCK and the CCK by use of the specified algorithm TA71. The algorithm shall be run whenever a new GCK is distributed or when a new CCK is issued (for instance caused by entering a new location area).

Input from ME: Record number in EF_{MGCK}, record number in EF_{CCK} to be used.

Input from EF: GCK, CCK.

Output to EF: MGCK (to EF_{MGCK}).

Output to ME: None.

9 Description of the commands

This clause states the general principles for mapping the functions described in clause 8 onto Application Protocol Data Units (APDU) which are used by the transmission protocol.

9.1 Mapping principles

An APDU can be a command APDU or a response APDU.

A command APDU has the following general format:

CLA	INS	P1	P2	P3	Data

The response APDU has the following general format:

Data	SW1	SW2

An APDU is transported by the T = 0 transmission protocol without any change. Other protocols might embed an APDU into their own transport structure (see ISO/IEC 7816-3 [21]).

The bytes have the following meaning:

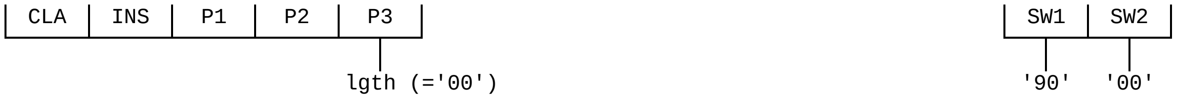
- CLA is the class of instruction (ISO/IEC 7816-3 [21]), 'A0' is used in the TETRA application;
- INS is the instruction code (ISO/IEC 7816-3 [21]) as defined in this clause for each command;
- P1, P2, P3 are parameters for the instruction. They are specified in table 4. 'FF' is a valid value for P1, P2 and P3. P3 gives the length of the data element. P3 = '00' introduces a 256 byte data transfer from the SIM in an outgoing data transfer command (response direction). In an ingoing data transfer command (command direction), P3 = '00' introduces no transfer of data;
- SW1 and SW2 are the status words indicating the successful or unsuccessful outcome of the command.

For some of the functions described in clause 8 it is necessary for T = 0 to use a supplementary transport service command (GET RESPONSE) to obtain the output data. For example, the SELECT function needs the following two commands:

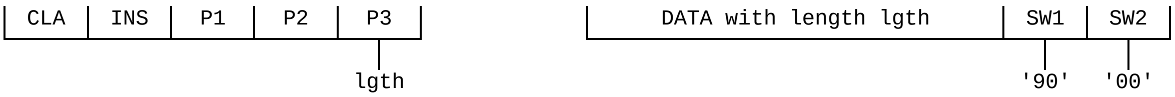
- the first command (SELECT) has both parameters and data serving as input for the function;
- the second command (GET RESPONSE) has a parameter indicating the length of the data to be returned.

If the length of the response data is not known beforehand, then its correct length may be obtained by applying the first command and interpreting the status words. SW1 shall be '9F' and SW2 shall give the total length of the data. Other status words may be present in case of an error. The various cases are:

Case 1: No input/No output

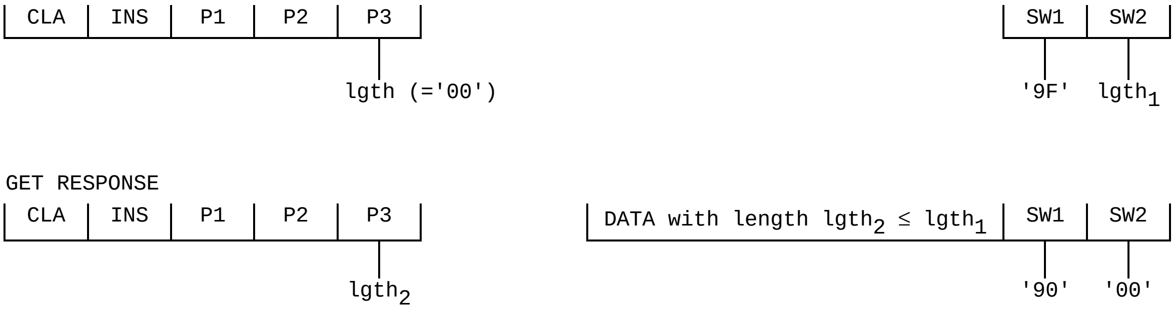


Case 2: No input/Output of known length

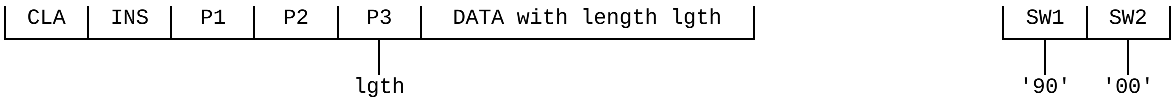


NOTE: lgth='00' causes a data transfer of 256 bytes.

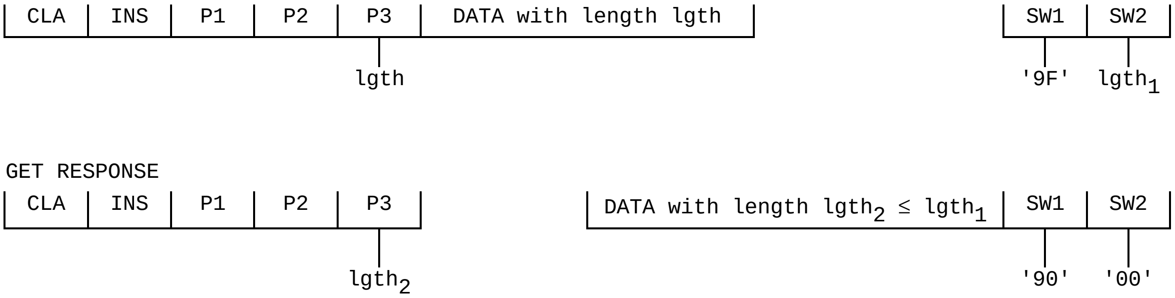
Case 3: No Input/Output of unknown length



Case 4: Input/No output



Case 5: Input/Output of known or unknown length



For cases 3 and 5, when Status Word 1/Status Word 2 (SW1/SW2) indicates there is response data (i.e. SW1/SW2 = '9FXX'), then, if the ME requires to get this response data, it shall send a GET RESPONSE command as described in the relevant case above.

If the TETRA application is one of several applications in a multi-application card, other commands with CLA not equal to 'A0' may be sent by the terminal. This shall not influence the state of the TETRA application.

9.2 Coding of the commands

Table 4 gives the coding of the commands. The direction of the data is indicated by (S) and (R), where (S) stands for data sent by the ME while (R) stands for data received by the ME. Offset is coded on 2 bytes where P1 gives the high order byte and P2 the low order byte. '00 00' means no offset and reading/updating starts with the first byte while an offset of '00 01' means that reading/updating starts with the second byte.

In addition to the instruction codes specified in table 4 the following codes are reserved:

TETRA operational phase:

'7X' with X even except for '76'

Administrative management phase:

'2E', '38, '3A', '3C', '3E'

Table 4: Coding of the commands

COMMAND	INS	P1	P2	P3	S/R
SELECT	'A4'	'00'	'00'	'02'	S/R
STATUS	'F2'	'00'	'00'	lgth	R
READ BINARY	'B0'	offset high	offset low	lgth	R
UPDATE BINARY	'D6'	offset high	offset low	lgth	S
READ RECORD	'B2'	rec No.	mode	lgth	R
UPDATE RECORD	'DC'	rec No.	mode	lgth	S
SEEK	'A2'	'00'	type/mode	lgth	S/R
READ KEY	'BE'	rec No.	'04'	lgth	R
VERIFY CHV	'20'	'00'	CHV No.	'08'	S
CHANGE CHV	'24'	'00'	CHV No.	'10'	S
DISABLE CHV	'26'	'00'	'01'	'08'	S
ENABLE CHV	'28'	'00'	'01'	'08'	S
UNBLOCK CHV	'2C'	'00'	see note	'10'	S
INVALIDATE	'04'	'00'	'00'	'00'	-
REHABILITATE	'44'	'00'	'00'	'00'	-
GET RANDOM	'CE'	'00'	'00'	'0A'	R
TA11/12 ALGORITHM	'40'	'00'	'00'	'14'	S/R
TA21/22 ALGORITHM	'42'	'00'	'00'	'0E'	S
TB4/TE ALGORITHM	'46'	'00'	'00'	'0A'	R
TA32 ALGORITHM	'48'	'00'	'00'	'11'	S
TA41/TA82 ALGORITHM	'4A'	rec No.	'00'	'17'	S
TA41/52 ALGORITHM	'4C'	'00'	'00'	'1B'	S
TA71 ALGORITHM	'4E'	target record no	input record no.	'0A'	-
GET RESPONSE	'C0'	'00'	'00'	lgth	R
NOTE: If the UNBLOCK CHV command applies to CHV1 then P2 is coded '00'; if it applies to CHV2 then P2 is coded '02'.					

Definitions and codings used in the response parameters/data of the commands are given in clause 9.2.23.

9.2.1 SELECT

COMMAND	CLASS	INS	P1	P2	P3
SELECT	'A0'	'A4'	'00'	'00'	'02'

Command parameters/data:

Byte(s)	Description	Length
1 - 2	File ID	2

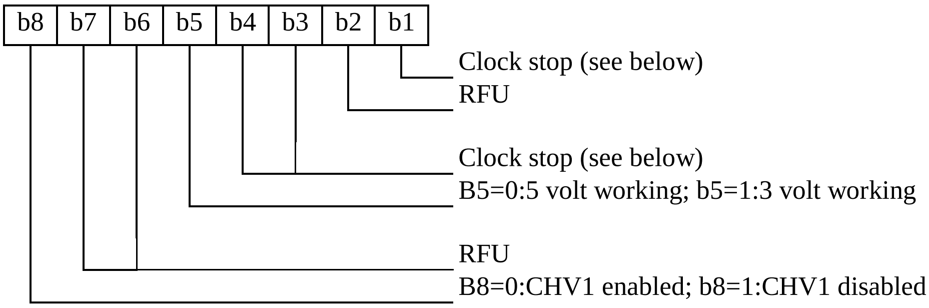
Response parameters/data in case of an MF or DF:

Byte(s)	Description	Length
1 - 2	RFU	2
3 - 4	Total amount of memory of the selected directory which is not allocated to any of the DFs or EFs under the selected directory	2
5 - 6	File ID	2
7	Type of file (see clause 9.3)	1
8 - 12	RFU	5
13	Length of the following data (byte 14 to the end)	1
14 - 34	TETRA specific data	21

TETRA specific data:

Byte(s)	Description	Length
14	File characteristics (see detail 1)	1
15	Number of DFs which are a direct child of the current directory	1
16	Number of EFs which are a direct child of the current directory	1
17	Number of CHVs, UNBLOCK CHVs and administrative codes	1
18	RFU	1
19	CHV1 status (see detail 2)	1
20	UNBLOCK CHV1 status (see detail 2)	1
21	CHV2 status (see detail 2)	1
22	UNBLOCK CHV2 status (see detail 2)	1
23	RFU	1
24 - 34	Reserved for the administrative management (optional)	0≤lgth≤11
NOTE 1: Byte 35 and following are RFU. NOTE 2: The STATUS information of the MF and DF$_{TETRA}$ provide some identical application specific data, e.g. CHV status. On a multi-application card the MF should not contain any application specific data. Such data is obtained by terminals from the specific application directories. ME manufacturers should take this into account and therefore not use application specific data which may exist in the MF of a mono-application SIM.		

Detail 1: File characteristics.


b8
b7
b6
b5
b4
b3
b2
b1
Clock stop (see below)
RFU
Clock stop (see below)
B5=0:5 volt working; b5=1:3 volt working
RFU
B8=0:CHV1 enabled; b8=1:CHV1 disabled

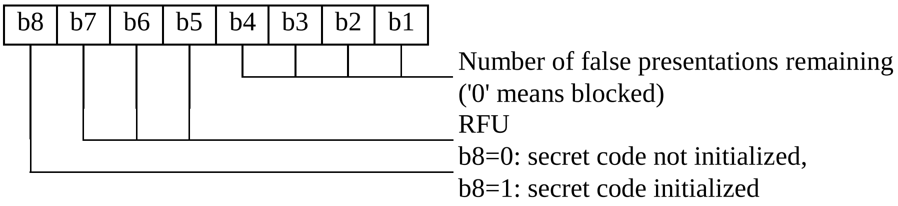
The coding of the conditions for stopping the clock is as follows:

Bit b1	Bit b3	Bit b4;	
1	0	0	clock stop allowed, no preferred level;
1	1	0	clock stop allowed, high level preferred;
1	0	1	clock stop allowed, low level preferred;
0	0	0	clock stop not allowed;
0	1	0	clock stop not allowed, unless at high level;
0	0	1	clock stop not allowed, unless at low level.

If bit b1 (column 1) is coded 1, stopping the clock is allowed at high or low level. In this case columns 2 (bit b3) and 3 (bit b4) give information about the preferred level (high or low, resp.) at which the clock may be stopped.

If bit b1 is coded 0, the clock may be stopped only if the mandatory condition in column 2 (b3 = 1, i.e. stop at high level) or column 3 (b4 = 1, i.e. stop at low level) is fulfilled. If all 3 bits are coded 0, then the clock shall not be stopped.

Detail 2: Status byte of a secret code.



Number of false presentations remaining ('0' means blocked)

RFU

b8=0: secret code not initialized,
b8=1: secret code initialized

Response parameters/data in case of an EF:

Byte(s)	Description	Length
1 - 2	RFU	2
3 - 4	File size (for transparent EF: the length of the body part of the EF) (for linear fixed, cyclic or key EF: record length multiplied by the number of records of the EF)	2
5 - 6	File ID	2
7	Type of file (see clause 9.3)	1
8	RFU	1
9 - 11	Access conditions (see clause 9.3)	3
12	File status (see clause 9.3)	1
13	Length of the following data (byte 14 to the end)	1
14	Structure of EF (see clause 9.3)	1
15	Length of a record (see detail 4)	1
NOTE: Byte 16 and following are RFU.		

Detail 4: Byte 15.

For cyclic, linear fixed and key EFs this byte denotes the length of a record. For a transparent EF, this byte shall be coded '00', if this byte is sent by the SIM.

9.2.2 STATUS

COMMAND	CLASS	INS	P1	P2	P3
STATUS	'A0'	'F2'	'00'	'00'	lgth

The response parameters/data are identical to the response parameters/data of the SELECT command in case of an MF or DF.

9.2.3 READ BINARY

COMMAND	CLASS	INS	P1	P2	P3
READ BINARY	'A0'	'B0'	offset high	offset low	lgth

Response parameters/data:

Byte(s)	Description	Length
1 - lgth	Data to be read	lgth

9.2.4 UPDATE BINARY

COMMAND	CLASS	INS	P1	P2	P3
UPDATE BINARY	'A0'	'D6'	offset high	offset low	lgth

Command parameters/data:

Byte(s)	Description	Length
1 - lgth	Data	lgth

9.2.5 READ RECORD

COMMAND	CLASS	INS	P1	P2	P3
READ RECORD	'A0'	'B2'	Rec.No.	Mode	lgth

Parameter P2 specifies the mode:

'02' = next record;

'03' = previous record;

'04' = absolute mode/current mode, the record number is given in P1 with P1 = '00' denoting the current record.

Response parameters/data:

Byte(s)	Description	Length
1 - lgth	The data of the record	lgth

9.2.6 UPDATE RECORD

COMMAND	CLASS	INS	P1	P2	P3
UPDATE RECORD	'A0'	'DC'	Rec.No.	Mode	lgth

Parameter P2 specifies the mode:

'02' = next record;

'03' = previous record;

'04' = absolute mode/current mode; the record number is given in P1 with P1 = '00' denoting the current record.

Command parameters/data:

Byte(s)	Description	Length
1 - lgth	Data	lgth

9.2.7 READ KEY

COMMAND	CLASS	INS	P1	P2	P3
READ RECORD	'A0'	'BE'	rec No.	'04'	lgth

Response parameters/data:

Byte(s)	Description	Length
1 - lgth	The data of the record	lgth

9.2.8 SEEK

COMMAND	CLASS	INS	P1	P2	P3
SEEK	'A0'	'A2'	'00'	Type/Mode	lgth

Parameter P2 specifies type and mode:

'x0' = from the beginning forward;

'x1' = from the end backward;

'x2' = from the next location forward;

'x3' = from the previous location backward;

with x = '0' specifies type 1 and x = '1' specifies type 2 of the SEEK command.

Command parameters/data:

Byte(s)	Description	Length
1 - lgth	Pattern	lgth

There are no response parameters/data for a type 1 SEEK. A type 2 SEEK returns the following response parameters/data:

Byte(s)	Description	Length
1	Record number	1

9.2.9 VERIFY CHV

COMMAND	CLASS	INS	P1	P2	P3
VERIFY CHV	'A0'	'20'	'00'	CHV No.	'08'

Parameter P2 specifies the CHV:

'01' = CHV1;

'02' = CHV2.

Command parameters/data:

Byte(s)	Description	Length
1 - 8	CHV value	8

9.2.10 CHANGE CHV

COMMAND	CLASS	INS	P1	P2	P3
CHANGE CHV	'A0'	'24'	'00'	CHV No.	'10'

Parameter P2 specifies the CHV:

'01' = CHV1;

'02' = CHV2.

Command parameters/data:

Byte(s)	Description	Length
1 - 8	Old CHV value	8
9 - 16	New CHV value	8

9.2.11 DISABLE CHV

COMMAND	CLASS	INS	P1	P2	P3
DISABLE CHV	'A0'	'26'	'00'	'01'	'08'

Command parameters/data:

Byte(s)	Description	Length
1 - 8	CHV1 value	8

9.2.12 ENABLE CHV

COMMAND	CLASS	INS	P1	P2	P3
ENABLE CHV	'A0'	'28'	'00'	'01'	'08'

Command parameters/data:

Byte(s)	Description	Length
1 - 8	CHV1 value	8

9.2.13 UNBLOCK CHV

COMMAND	CLASS	INS	P1	P2	P3
UNBLOCK CHV	'A0'	'2C'	'00'	CHV No.	'10'

Parameter P2 specifies the CHV:

00 = CHV1;

02 = CHV2.

NOTE: The coding '00' for CHV1 differs from the coding of CHV1 used for other commands.

Command parameters/data:

Byte(s)	Description	Length
1 - 8	UNBLOCK CHV value	8
9 - 16	New CHV value	8

9.2.14 INVALIDATE

COMMAND	CLASS	INS	P1	P2	P3
INVALIDATE	'A0'	'04'	'00'	'00'	'00'

9.2.15 REHABILITATE

COMMAND	CLASS	INS	P1	P2	P3
REHABILITATE	'A0'	'44'	'00'	'00'	'00'

9.2.16 GET RANDOM

COMMAND	CLASS	INS	P1	P2	P3
GET RANDOM	'A0'	'CE'	'00'	'00'	'0A'

Response parameters/data:

Byte(s)	Description	Length
1 - 10	RAND2	10

9.2.17 TA11/TA12 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA11/TA12 ALGORITHM	'A0'	'40'	'00'	'00'	'14'

Command parameters/data:

Byte(s)	Description	Length
1 - 10	RAND1	10
11 - 20	RS	10

Response parameters/data:

Byte(s)	Description	Length
4	RES1	4

See EN 300 392-7 [3] for use of RES1 and for size of the cryptographic parameters.

9.2.18 TA21/TA22 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA21/TA22 ALGORITHM	'A0'	'42'	'00'	'00'	'0E'

Command parameters/data:

Byte(s)	Description	Length
1 - 4	RES2	4
5 - 14	RS	10

Response parameters/data:

Byte(s)	Description	Length
4	XRES2	4

9.2.19 TB4/TE ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TB4 ALGORITHM	'A0'	'46'	'00'	'00'	'0A'

Response parameters/data:

Byte(s)	Description	Length
1 - 10	DCK	10

9.2.20 TA32 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA32 ALGORITHM	'A0'	'48'	'00'	'00'	'11'

Command parameters/data:

Byte(s)	Description	Length
1 - 15	SCCK	15
16 - 17	CCK-id	2

9.2.21 TA41/TA82 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA41/TA82 ALGORITHM	'A0'	'4A'	'Rec.No.'	'00'	'17'

P1 specifies the target record number of the record within the EF_{GCK}.

Command parameters/data:

Byte(s)	Description	Length
1 - 6	GTSI	6
7 - 21	SGCK	15
22 - 23	GCK-VN	2

9.2.22 TA41/TA52 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA41/TA52 ALGORITHM	'A0'	'4C'	'00'	'00'	'1B'

Command parameters/data:

Byte(s)	Description	Length
1 - 15	SSCK	15
16 - 17	SCK-VN	2
18 - 27	RSO	10

9.2.23 TA71 ALGORITHM

COMMAND	CLASS	INS	P1	P2	P3
TA71 ALGORITHM	'A0'	'4E'	target record	input record	'00'

Parameter P1 specifies the target record number in EF_{MGCK}:

P1 = '00' Update all MGCKs;

P1 ≠ '00' Parameter P1 gives the record number to be updated.

Parameter P2 specifies the record number (1 or 2) in EF_{CCK} from which the CCK shall be retrieved:

P2 = '01' or '02' and denotes the record number in EF_{CCK}.

9.2.24 GET RESPONSE

COMMAND	CLASS	INS	P1	P2	P3
GET RESPONSE	'A0'	'C0'	'00'	'00'	lgth

The response data depends on the preceding command. Response data is available after the commands TA11/12 ALGORITHM, SEEK (type 2) and SELECT. If the command GET RESPONSE is executed, it is required that it is executed immediately after the command it is related to (no other command shall come between the command/response pair and the command GET RESPONSE). If the sequence is not respected, the SIM shall send the status information "technical problem with no diagnostic given" as a reaction to the GET RESPONSE.

Since the MF is implicitly selected after activation of the SIM, GET RESPONSE is also allowed as the first command after activation.

The response data itself is defined in the clause for the corresponding command.

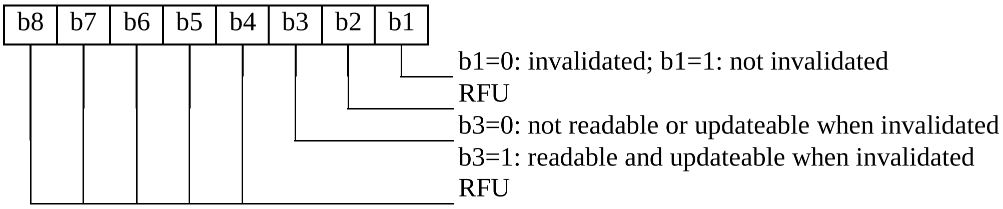
9.3 Definitions and coding

The following definitions and coding are used in the response parameters/data of the commands.

Coding: Each byte is represented by bits b8 to b1, where b8 is the Most Significant Bit (MSB) and b1 is the Least Significant Bit (LSB). In each representation the leftmost bit is the MSB.

RFU: In a TETRA specific card all bytes which are RFU shall be set to '00' and RFU bits to 0. Where the TETRA application exists on a multi-application card or is built on a generic telecommunications card (e.g. TE9) then other values may apply. The values will be defined in the appropriate specifications for such cards. These bytes and bits shall not be interpreted by an ME in a TETRA session.

File status:


b8
b7
b6
b5
b4
b3
b2
b1
b1=0: invalidated; b1=1: not invalidated
RFU
b3=0: not readable or updateable when invalidated
b3=1: readable and updateable when invalidated
RFU

Bit b3 may be set to 1 in special circumstances when it is required that the EF can be read and updated even if the EF is invalidated, e.g. reading and updating the EF_{ADN} when the Fixed Dialling Number (FDN) feature is enabled.

Structure of file:

'00'transparent;

'01'linear fixed;

'03'cyclic;

'11'key.

Type of file:

'00'RFU;

'01'MF;

'02'DF;

'04'EF.

Coding of CHVs and UNBLOCK CHVs:

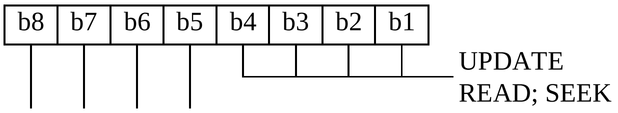
A CHV is coded on 8 bytes. Only (decimal) digits (0-9) shall be used, coded in ITU-T Recommendation T.50 [26] with bit 8 set to zero. The minimum number of digits is 4. If the number of digits presented by the user is less than 8 then the ME shall pad the presented CHV with 'FF' before sending it to the SIM.

The coding of the UNBLOCK CHVs is identical to the coding of the CHVs. However, the number of (decimal) digits is always 8.

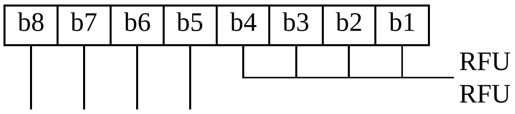
Coding of access conditions:

The access conditions for the commands are coded on bytes 9,10 and 11 of the response data of the SELECT command. Each condition is coded on 4 bits as shown in table 2.

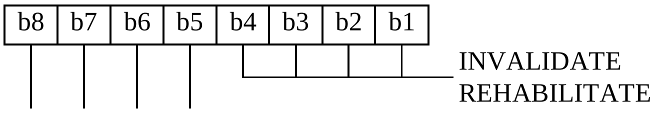
Byte 9:



Byte 10:



Byte 11:



9.4 Status conditions returned by the card

This clause specifies the coding of the status words SW1 and SW2.

9.4.1 Responses to commands which are correctly executed

SW1	SW2	Description
'90'	'00'	normal ending of the command
'9F'	'XX'	length 'XX' of the response data

9.4.2 Memory management

SW1	SW2	Error description
'92'	'0X'	command successful but after using an internal update retry routine 'X' times
'92'	'40'	memory problem

9.4.3 Referencing management

SW1	SW2	Error description
'94'	'00'	no EF selected
'94'	'02'	out of range (invalid address)
'94'	'04'	file ID not found pattern not found
'94'	'08'	file is inconsistent with the command

9.4.4 Security management

SW1	SW2	Error description
'98'	'02'	no CHV initialized
'98'	'04'	access condition not fulfilled unsuccessful CHV verification, at least one attempt left unsuccessful UNBLOCK CHV verification, at least one attempt left
'98'	'08'	in contradiction with CHV status
'98'	'10'	in contradiction with invalidation status
'98'	'40'	unsuccessful CHV verification, no attempt left unsuccessful UNBLOCK CHV verification, no attempt left CHV blocked UNBLOCK CHV blocked
'98'	'60'	manipulation flag set
'98'	'70'	SwMI authentication unsuccessful

9.4.5 Application independent errors

SW1	SW2	Error description
'67'	'XX'	incorrect parameter P3 (see note 3)
'6B'	'XX' (note 1)	incorrect parameter P1 or P2 (see note 2)
'6D'	'XX' (note 1)	unknown instruction code given in the command
'6E'	'XX' (note 1)	wrong instruction class given in the command
'6F'	'XX' (note 1)	technical problem with no diagnostic given
NOTE 1: These values of 'XX' are specified by ISO/IEC; at present the default value 'XX' = '00' is the only one defined. NOTE 2: When the error in P1 or P2 is caused by the addressed record being out of range, then the return code '94 02' shall be used. NOTE 3: 'XX' gives the correct length or states that no additional information is given ('XX' = '00').		

9.4.6 Commands versus possible status responses

Table 5 shows for each command the possible status conditions returned (marked by an asterisk *).

Table 5: Commands and status words

	OK		Mem Status		Refer.Status				Security status							Application Independent Errors				
Commands	**9000**	**9FXX**	**920X**	**9240**	**9400**	**9402**	**9404**	**9408**	**9802**	**9804**	**9808**	**9810**	**9840**	**9860**	**9870**	**67XX**	**6BXX**	**6DXX**	**6EXX**	**6FXX**
Select		*		*			*									*	*		*	*
Status	*			*												*	*		*	*
Update Binary	*		*	*	*			*		*		*				*	*		*	*
Update Record	*		*	*	*	*		*		*		*				*	*		*	*
Read Binary	*			*	*			*		*		*				*	*		*	*
Read Record	*			*	*	*		*		*		*				*	*		*	*
Read Key	*			*	*	*		*		*		*				*	*		*	*
Seek	*	*		*	*		*	*		*		*				*	*		*	*
Verify CHV	*		*	*					*	*	*		*			*	*		*	*
Change CHV	*		*	*					*	*	*		*			*	*		*	*
Disable CHV	*		*	*					*	*	*		*			*	*		*	*
Enable CHV	*		*	*					*	*	*		*			*	*		*	*
Unblock CHV	*		*	*					*	*	*		*			*	*		*	*
Invalidate	*		*	*	*					*		*				*	*		*	*
Rehabilitate	*		*	*	*					*		*				*	*		*	*
Get Random	*			*												*	*		*	*
TA11/TA12 Algorithm		*		*						*						*	*		*	*
TA21/TA22 Algorithm	*			*						*					*	*	*		*	*
TB4/TE Algorithm	*			*						*					*	*	*		*	*
TA32 Algorithm	*			*						*				*		*	*		*	*
TA41/TA82 Algorithm	*			*						*				*		*	*		*	*
TA41/TA52 Algorithm	*			*						*				*		*	*		*	*
TA71 Algorithm	*			*						*				*		*	*		*	*
Get Response	*			*												*	*		*	*

10 Contents of the EFs

This clause specifies the EFs for the TETRA session defining access conditions, data items and coding. A data item is a part of an EF which represents a complete logical entity, e.g. the alpha tag in a EF_{ADN} record.

EFs or data items having an unassigned value, or, which during the TETRA session, are cleared by the ME, shall have their bytes set to 'FF'. After the administrative phase all data items shall have a defined value or have their bytes set to 'FF'. If a data item is "deleted" during a TETRA session by the allocation of a value specified in another TETRA TS, then this value shall be used, and the data item is not unassigned.

EFs are mandatory (M) or optional (O). The file size of an optional EF may be zero. All implemented EFs with a file size greater than zero shall contain all mandatory data items. Optional data items may either be filled with 'F', or, if located at the end of an EF, need not exist.

Using the command GET RESPONSE the ME can determine the length of variable length records (e.g. 1 to X).

NOTE: The field "Update activity" has only meaning to the card manufacturer to help choosing proper memory management for EFs. If an EF is updated very seldom, e.g. once during the administrative phase, it is set to "low". If an EF is updated or may be updated in every TETRA session it is set to "high". The actual update activity of certain EFs also depends on the system. Therefore the update activity of an EF is set to high if it may be updated frequently in some systems. For example, high security systems may want to update cipher keys frequently, but less secure systems may update keys only when a particular reason to do it arises.

10.1 Void

10.2 Contents of the EFs at the MF level

There are three EFs at the MF level.

10.2.1 EF_{ICCID} (Card Identification)

This EF provides a unique identification number for the SIM.

<table>
<tr><td colspan="2">Identifier: '2FE2'</td><td colspan="2">Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="3">File size: 10 bytes</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALW
UPDATE NEV
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1 - 10</td><td colspan="2">Identification number</td><td>M</td><td>10</td></tr>
</table>

Identification number:

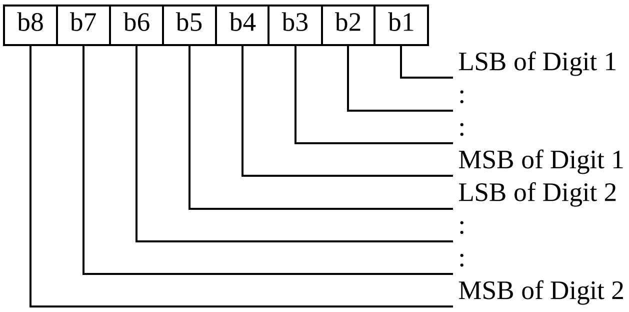
Contents:

Card identification number according to ITU-T Recommendation E.118 [27].

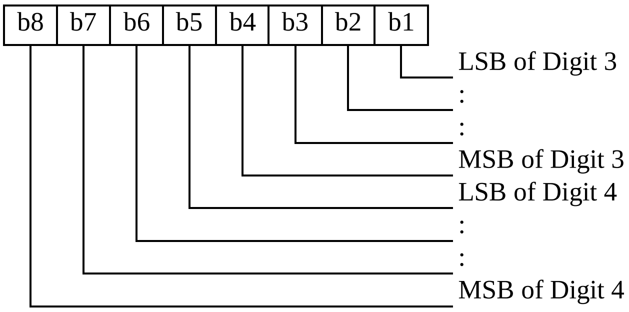
Coding:

Binary Coded Decimal (BCD), left justified and padded with 'F'.

Byte 1:



Byte 2:



etc.

10.2.2 EF$_{DIR}$ (Application directory)

An EF containing a list of applications supported by the card, and optional related data elements defined in ISO 7816-5 [22].

Identifier: '2F00 '	Structure: transparent	Optional
File size: X bytes	Update activity: low	

Access Conditions:

READ	ALW/CHV1 (see note 1)
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Application identifier tag ('4F')	M	1
2	Application identifier length	M	1
3	Application identifier (see note 2)	M	1-16
	Application label tag ('50')	M	1
	Application label length	M	1
	Application label (verbal description)	M	0-16
	Path tag ('51')	M	1
	Path length	M	1
	Path	M	X
	Second application information		

NOTE 1: Access conditions to this file are defined during the administrative phase.

NOTE 2: Application identifiers are allocated by ETSI.

10.2.3 EF$_{LP}$ (Language Preference)

This EF contains the codes for one or more languages. This information, determined by the user/operator, defines the preferred languages of the user in order of priority. This information may be used by the ME for MMI purposes and for short message handling.

Identifier: '2F05 '	Structure: transparent	Mandatory
File size: 2n bytes	Update activity: low	

Access Conditions:

READ	ALW
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 - 2	1st language code (highest priority)	M	2
3 - 4	2nd language code	O	2
2n-1 – 2n	nth language code (lowest priority)	O	2

Coding: As defined in ISO 639 [23]

Using the command GET RESPONSE, the ME can determine the size of the EF.

10.3 Contents of the EFs at the TETRA application level

10.3.1 EF$_{SST}$ (SIM Service Table)

The purpose of this EF is to indicate which of the optional services and EFs are available.

NOTE 1: Having the presence of optional services indicated simplifies their handling for the ME.

<table>
<tr><td colspan="2">Identifier: '6F01'</td><td colspan="2">Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="3">File size: X bytes, X ≥ 4</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ CHV1
UPDATE ADM
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td colspan="2">Services n° 1 to n° 8</td><td>M</td><td>1</td></tr>
<tr><td>2</td><td colspan="2">Services n° 9 to n° 16</td><td>M</td><td>1</td></tr>
<tr><td>3</td><td colspan="2">Services n° 17 to n° 24</td><td>M</td><td>1</td></tr>
<tr><td>4</td><td colspan="2">Services n° 25 to n° 32</td><td>M</td><td>1</td></tr>
<tr><td>5</td><td colspan="2">Services n^{o} 33 to n^{o}4 0</td><td>M</td><td>1</td></tr>
<tr><td>6</td><td colspan="2">Services n^{o} 41 to n^{o} 48</td><td>M</td><td>1</td></tr>
<tr><td>etc.</td><td colspan="2"></td><td></td><td></td></tr>
<tr><td>X</td><td colspan="2">Service (8X-7) to (8X)</td><td>O</td><td>1</td></tr>
</table>

Services:

Contents:

Service no. 1: CHV1 disable function

Service no. 2: ADNTETRA (Internal TETRA Phone Book) and Extension A

Service no. 3: ADNGWT (External phones), Gateway Extension 1 and Gateway table

Service no. 4: FDNTETRA and Extension B

Service no. 5: FDNGWT, Gateway Extension 2 and Gateway table

Service no. 6: SDNTETRA

Service no. 7: SDNGWT, Gateway Extension 3 and Gateway table

Service no. 8: LNDTETRA and Extension A

Service no. 9: LNDGWT, Gateway Extension 1 and Gateway table

Service no. 10: RFU

Service no. 11: CCK and CCK location areas

Service no. 12: SCK

Service no. 13: GCK and MGCK

Service no. 14: Service Provider Name

Service no. 15: Preferred Networks

Service no. 16: Username

Service no. 17: Authentication

Service no. 18: OTAR

Service no. 19: RFU

Service no. 20: Enhanced SIM-ME security

Service no. 21: RFU

Service no. 22: Status message texts

Service no. 23: SDS1 message texts

Service no. 24: SDS 123 Storage

Service no. 25: SDS 4 Storage (including the SDS 4 message storage status)

Service no. 26: Call Modifiers

Service no. 27: DMO channel information and MS allocation of DMO channels

Service no. 28: List of key holders

Service no. 29: DMO repeater and gateway list

Service no. 30: SDS Parameters Service no.31: Default Status Target

Service no. 32: SDS Delivery Report

Service no. 33: RFU

Service no. 34: Preferred Location Area

Service no. 35: Welcome Message

Service no. 36: ADN (External phones), Extension 1 and Gateway table

Service no. 37: FDN, Extension 2 and Gateway table

Service no. 38: SDN, Extension 3 and Gateway table

Service no. 39: LND, Extension 1 and Gateway table

Service no. 40: LNDComp

Service no. 41: Private Number information

Service no. 42: APN table

NOTE 2: Other services are possible in the future and will be coded on further bytes in the EF.

The coding falls under the responsibility of ETSI.

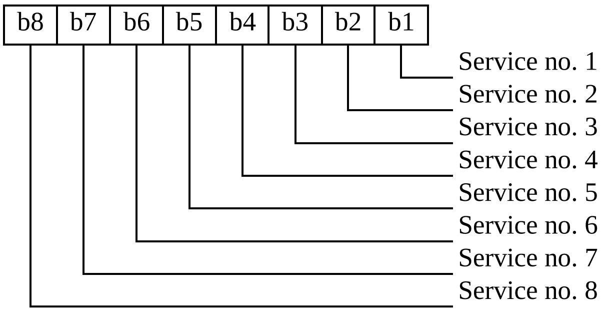
Coding:

1 bit is used to code each service:

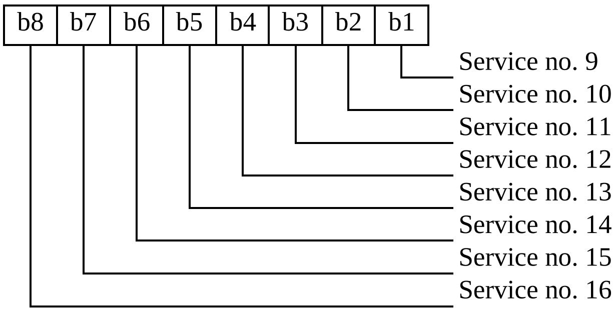
bit = 1: service available

bit = 0: service not available

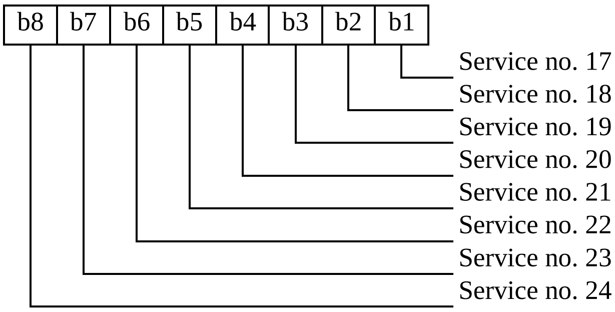
Byte 1:



Byte 2:

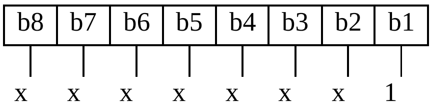


Byte 3:



etc.

The following example of coding for the first byte means that service no.1 "CHV1-Disabling" is available.



10.3.2 EF$_{ITSI}$ (Individual Tetra Subscriber Identity)

This EF contains the Individual Tetra Subscriber Identity number (ITSI). This EF shall not readable or updateable when invalidated.

Identifier: '6F02'	Structure: transparent		Mandatory
File size: 6 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE AUTI(see note) REHABILITATE NEV			
Bytes	Description	M/O	Length
1 - 6	ITSI	M	6
NOTE: This access condition is dependent on the security class of the Network. If mutual authentication is not supported by the SwMI then the access condition shall be ADM			

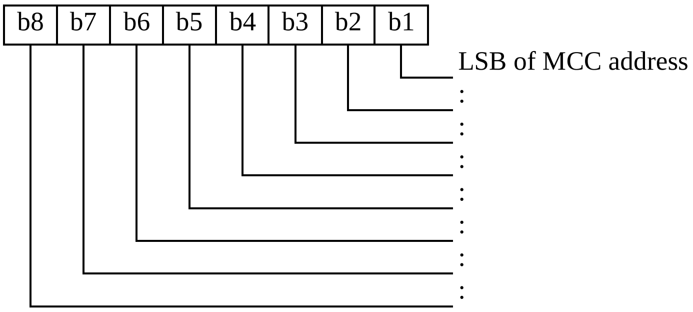
ITSI:

Contents:

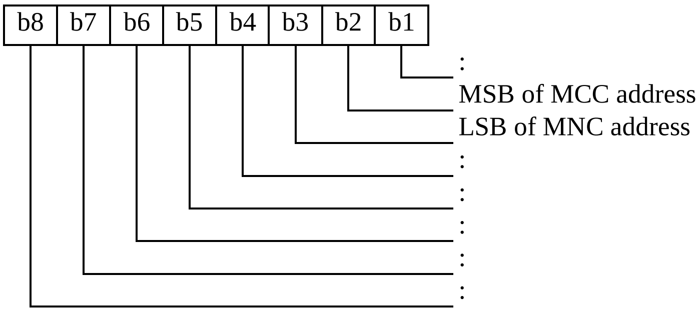
ITSI consists of Mobile Country Code (MCC), Mobile Network Code (MNC) and Individual Short Subscriber Identity (ISSI).

Coding:

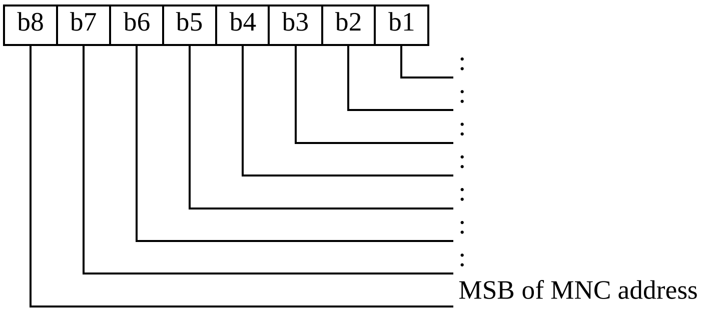
Byte 1:



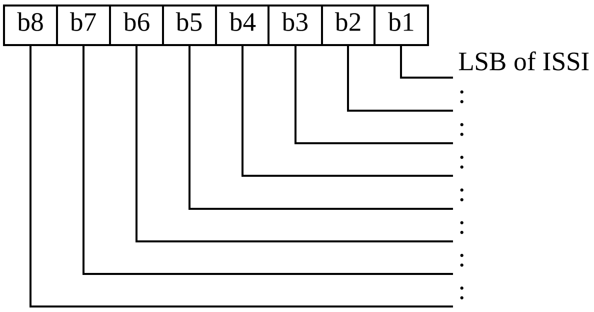
Byte 2:



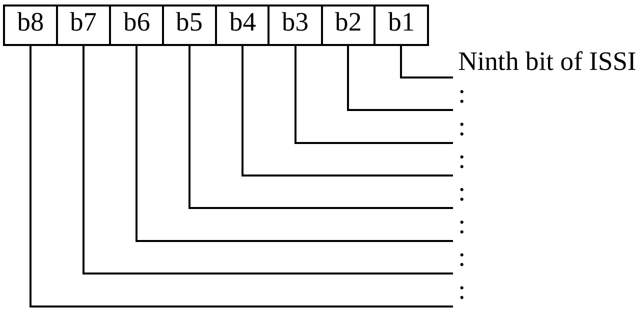
Byte 3:



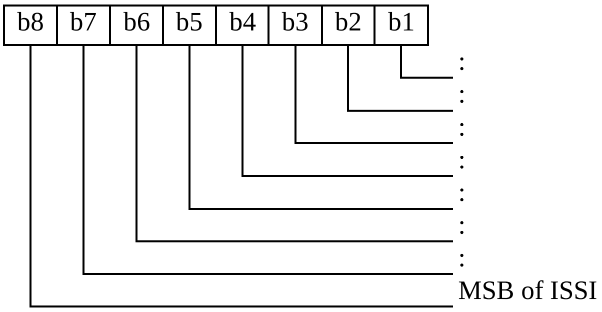
Byte 4:



Byte 5:



Byte 6:



NOTE: The network address of the ITSI shall be used as preferred network address.

10.3.3 $\text{EF}_{\text{ITSIDIS}}$ (ITSI Disabled)

This EF indicates if the ITSI is temporarily disabled.

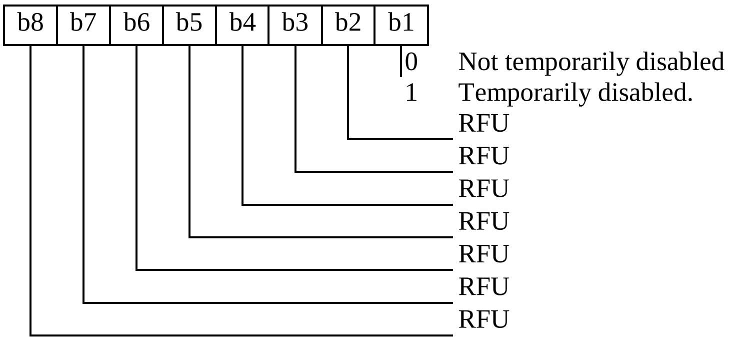
<table>
<tr><td colspan="2">Identifier: '6F03 '</td><td>Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="2">File size: 1 byte</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="4">Access Conditions:
READ CHV1
UPDATE AUTI(see note)
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td>Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td>Status</td><td>M</td><td>1</td></tr>
<tr><td colspan="4">NOTE: This access condition is dependent on the security class of the Network. If mutual authentication is not supported by the SwMI then the access condition shall be ADM</td></tr>
</table>

Status:

Contents:

The status bit indicates the temporary disable status of ITSI.

Coding:.



10.3.4 EF_{UNAME} (Username)

This EF contains the alphanumeric name corresponding to the ITSI.

<table>
<tr><td colspan="2">Identifier: '6F04 '</td><td>Structure: transparent</td><td>Optional</td></tr>
<tr><td colspan="2">File size: 20 bytes</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="4">Access Conditions:
READ CHV1
UPDATE ADM
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td>Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1-20</td><td>Name</td><td>M</td><td>20</td></tr>
</table>

Name:

Contents: The common name of the card holder to be displayed.

Coding: According to the default 8-bit alphabet ISO 8859-1 [24]. Unused bytes shall be set as 'FF'.

10.3.5 EF$_{SCT}$ (Subscriber Class Table)

This EF records the subscriber class membership of the ITSI subscription. The subscriber class membership shall be defined at subscription. The subscriber class element is used to subdivide the MS population in up to 16 classes.

The ITSI subscriber class may only be changed via the MMI by an authorized administrator or via the SwMI by the Network Operator or authorized system manager.

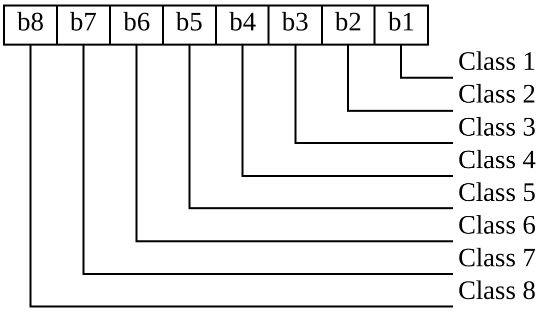
Identifier: '6F05 '	Structure: transparent		Mandatory
File size: 4 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Classes from 1 to 8	M	1
2	Classes from 9 to 16	M	1
3-4	Energy saving information	O	2

Classes from 1 to 8:

Contents: Indicates the class membership for classes from 1 to 8.

Coding: Bit value 1 means that user is a member, value 0 that user is not a member.

Byte 1:

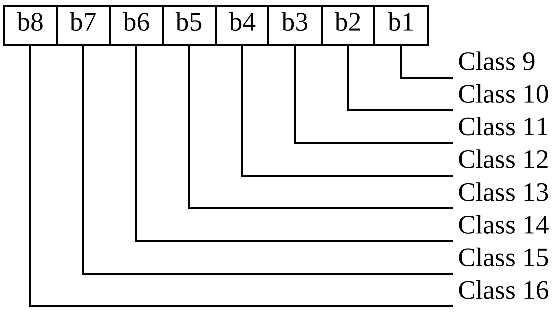


Classes from 9 to 16:

Contents: Indicates the class membership for classes from 9 to 16.

Coding: Bit value 1 means that user is a member, value 0 that user is not a member.

Byte 2:



Energy Saving Information:

Contents: Indicates which energy saving scheme (if any) is in operation and the starting point of the energy economy mode.

Coding: As per EN 300 392-2 [2](14 bits) with b8 and b7 of first byte RFU.

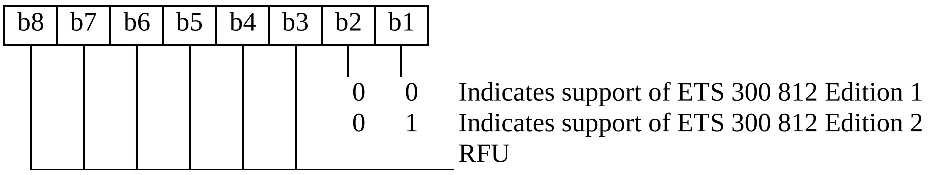
10.3.6 EF$_{PHASE}$ (Phase identification)

This EF contains information concerning the phase of the SIM.

<table>
<tr><td colspan="2">Identifier: '6F06 '</td><td colspan="2">Structure: transparent</td><td>Mandatory</td></tr>
<tr><td colspan="3">File size: 1 byte</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ ALW
UPDATE ADM
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1</td><td colspan="2">SIM Phase</td><td>M</td><td>1 byte</td></tr>
</table>

SIM Phase:

Coding:



0 0 Indicates support of ETS 300 812 Edition 1
0 1 Indicates support of ETS 300 812 Edition 2
RFU

All other codings are reserved for specification by ETSI

10.3.7 EF$_{CCK}$ (Common Cipher Key)

This EF shall contain 2 records.

<table>
<tr><td colspan="2">Identifier: '6F07 '</td><td colspan="2">Structure: key</td><td>Optional</td></tr>
<tr><td colspan="3">Record size: 12 bytes</td><td colspan="2">Update activity: high</td></tr>
<tr><td colspan="5">Access Conditions:
READ CHV1 (see note 1)
UPDATE NEV (see note 2)
INVALIDATE NEV
REHABILITATE NEV</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1-2</td><td colspan="2">CCK-id</td><td>M</td><td>2</td></tr>
<tr><td>3-12</td><td colspan="2">Common cipher key CCK</td><td>M</td><td>10</td></tr>
</table>

NOTE 1: Read access to this file is only possible by use of Read Key command.

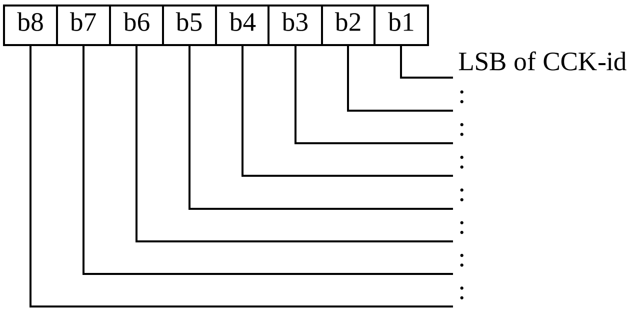
NOTE 2: This EF is updated using the TA32 algorithm on the SIM.

CCK-id:

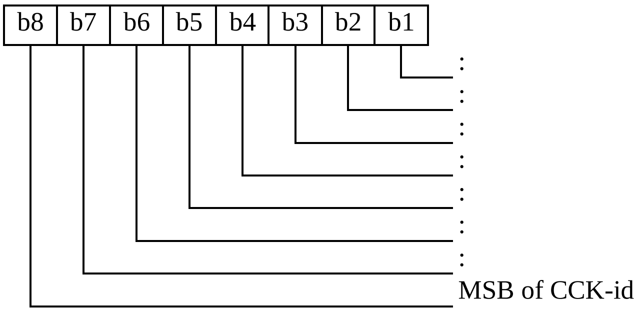
Contents: Common cipher key identity.

Coding:

Byte 1:



Byte 2:

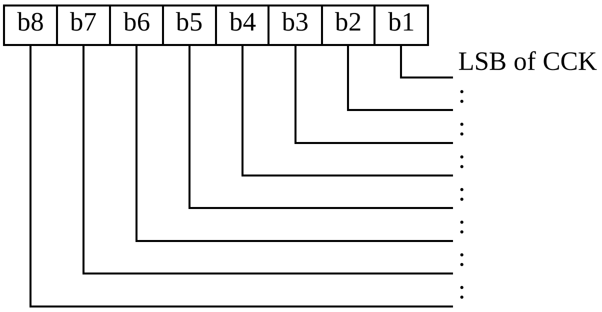


Common Cipher Key (CCK):

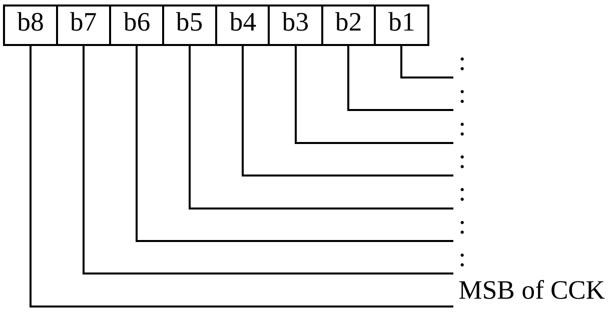
Contents: CCK.

Coding: CCK is coded in 10 bytes according to the following diagram:

Byte 3:



Byte 12:



10.3.8 EF$_{CCKLOC}$ (CCK location areas)

This EF defines the location area(s) the CCK is valid. If no location areas are defined the CCK is valid in the whole system.

Identifier: '6F08 '	Structure: transparent		Optional
File size: 31 bytes		Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Number of location areas	M	1
2-31	Location area	O	30

Number of location areas:

Contents: indicates the number location area elements there are to follow in 'Location area'.

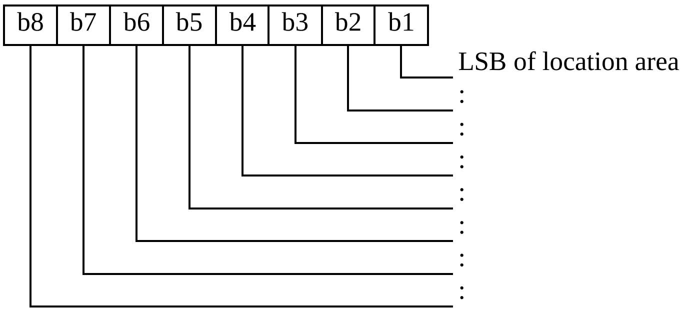
Coding: binary coded from 0 to 15. If value is 0, the CCK is valid system wide (see also EN 300 392-7 [3]).

Location area:

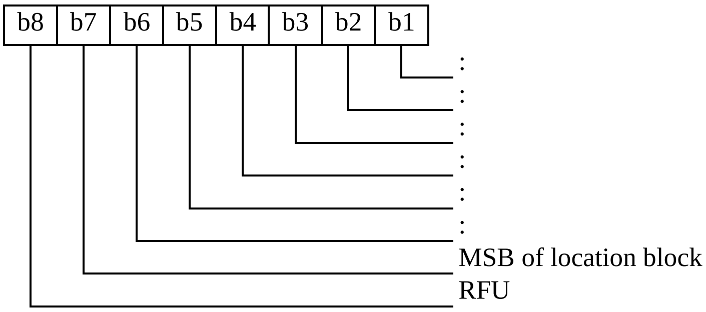
Contents: a list of location areas where CCKs are valid.

Coding: Each element is coded in 2 bytes, 14 bits. The first element (bytes 2 and 3) is shown below. See also EN 300 392-7 [3].

Byte 2:



Byte 3:



10.3.9 EF$_{SCK}$ (Static Cipher Keys)

This EF shall contain up to 32 records.

<table>
<tr><td colspan="2">Identifier: '6F09 '</td><td>Structure: key</td><td>Optional</td></tr>
<tr><td colspan="2">Record length: 12 bytes</td><td colspan="2">Update activity: high</td></tr>
<tr><td colspan="4">Access Conditions:
READ CHV1 (see note 1)
UPDATE NEV (see note 2)
INVALIDATE NEV
REHABILITATE NEV</td></tr>
<tr><td>Bytes</td><td>Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1-2</td><td>Static Cipher Key Version Number</td><td>M</td><td>2</td></tr>
<tr><td>3-12</td><td>Static Cipher Key</td><td>M</td><td>10</td></tr>
</table>

NOTE 1: Read access to this file is only possible by use of Read Key command.

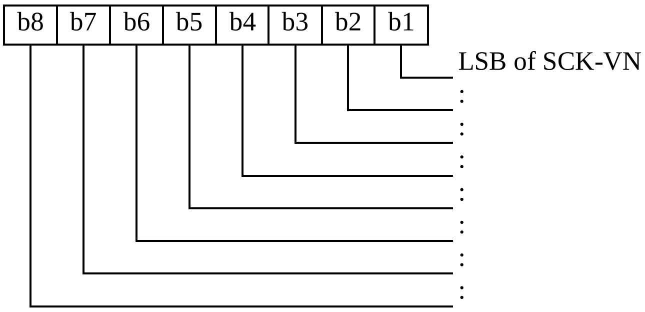
NOTE 2: This EF is updated using the TA41/52 algorithms on the SIM.

Static Cipher Key Version Number:

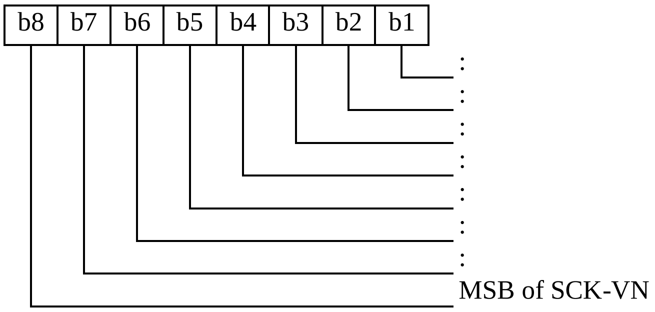
Contents: The Static Cipher Key Version Number.

Coding: The Static Cipher Key Version Number is coded according to the following diagram:

Byte 1:



Byte 2:

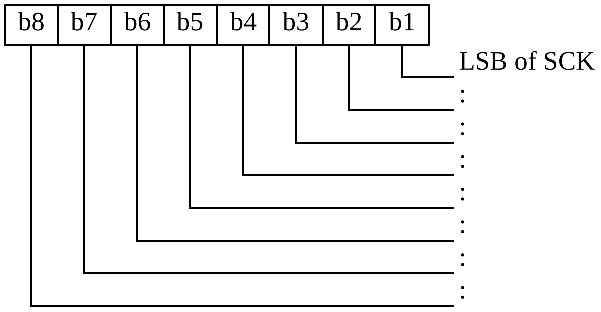


Static Cipher Key:

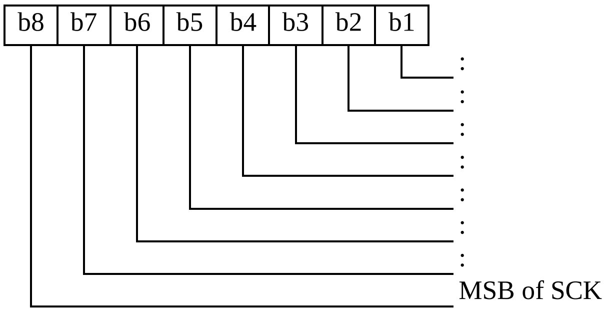
Contents: The Static Cipher Key.

Coding: The Static Cipher Key is coded in 10 bytes according to the following diagram:

Byte 3:



Byte 12:



10.3.10 EF$_{GSSIS}$ (Static GSSIs)

This EF contains the pre-programmed (by the operator or organization) group identities.

NOTE 1: Suggested number of static groups is between 1 and 10.

NOTE 2: Static GSSIs can not be updated after the administrative phase.

Identifier: '6F0A'	Structure: linear fixed	Mandatory
Record length: X + 4 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to X	Group name	M	X
X + 1	Network address record number	M	1
X + 2-X + 4	Group Identity (GSSI)	M	3

Group name:

Contents: Alphanumeric names for the static groups stored on the SIM.

Coding: The value of X may range from zero to 251.

Network address record number:

Contents: Record number of the corresponding network address. Network addresses are stored in EF$_{NWT}$.

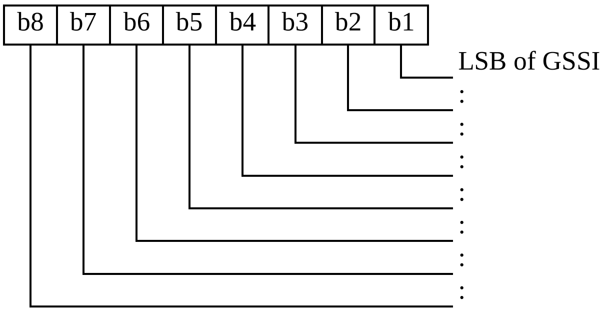
Coding: binary. Free records are indicated by NULL value ('00').

Group Identity (GSSI):

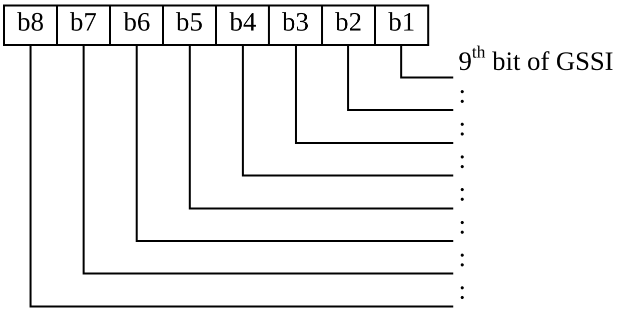
Contents: The short subscriber identity for the group.

Coding: Length of the GSSI is 24 bits.

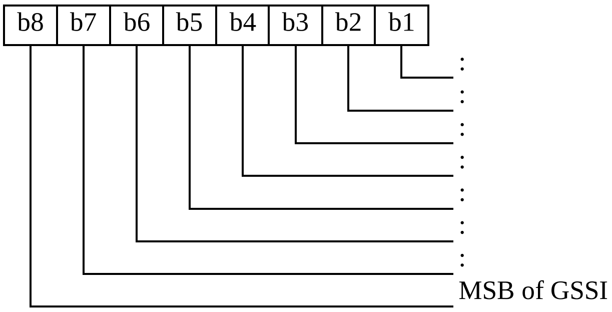
Byte X+3:



Byte X+4:



Byte X+5:



10.3.11 EF$_{GRDS}$ (Group related data for static GSSIs)

This EF contains information related to each static GSSI. There shall be a 1:1 relationship between each record in EF$_{GRDS}$ and the corresponding record in EF$_{GSSIS}$.

Identifier: '6F0B '	Structure: linear fixed		Mandatory
Record size: 2 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Key record number	M	1
2	Group related data	M	1

Key record number:

Contents: Class 2 systems record number of the corresponding SCK in the EF$_{SCK}$-file.

Contents: Class 3 systems record number of the corresponding GCK in the EF$_{GCK}$-file.

Coding: binary. In class 2 systems if there is no SCK defined for this group, key record number shall be NULL value ('00').

Coding: binary. In class 3 systems if there is no GCK defined for this group, key record number shall be NULL value ('00').

Group related data:

Contents:

Group Identity lifetime(2 bits): Shall indicate the attachment lifetime of the group identity. (EN 300 392-2 [2] clause 16.10.16)

Class of usage (3 bits). Shall indicate the importance of the group for the user and define the participation rules for the groups defined with Class of usage. (EN 300 392-2 [2] and EN 300 392-12-22 [4]).

Permanent Detachment Flag (1 bit). Shall indicate that whether a group identity was permanent detached by the SwMI.

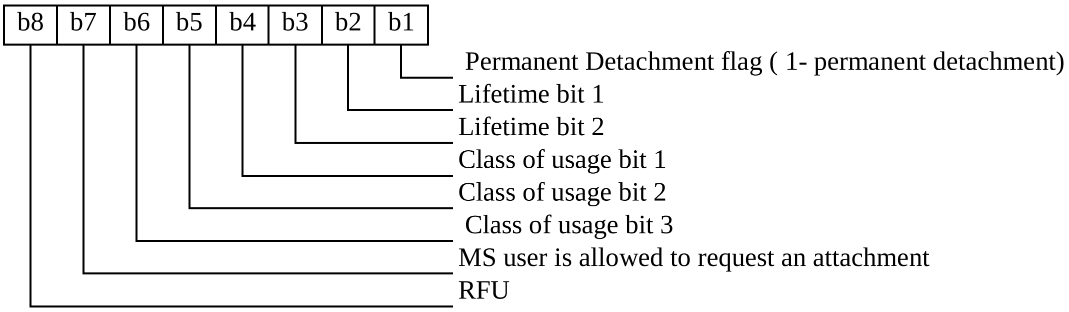
MS user is allowed to request an attachment (1 bit): Shall indicate whether MS user is allowed to request an attachment.

Table 6: Group identity attachment lifetime

Information element	Length	Value	Remark
Group Identity Lifetime	2	00	attachment not needed
		01	attachment for next ITSI attach required
		10	attachment not allowed for next ITSI attach
		11	attachment for next location update required

Coding:

Byte 2:



10.3.12 EF$_{GSSID}$ (Dynamic GSSIs)

This EF contains the dynamic group identities.

Identifier: '6F0C '	Structure: linear fixed	Mandatory
Record length: X + 4 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE AUTI(see note) INVALIDATE AUTI(see note) REHABILITATE NEV		

Bytes	Description	M/O	Length
1 to X	Group name	M	X
X + 1	Network address record number	M	1
X + 2-X + 4	Group Identity (GSSI)	M	3
NOTE: This access condition is dependent on the security class of the Network. If mutual authentication is not supported by the SwMI then the access condition shall be ADM			

See EF$_{GSSIS}$ (Static GSSIs) for contents and coding.

10.3.13 EF_{GRDD} (Group related data for dynamic GSSIs)

This EF contains information related to each dynamic GSSI. There shall be a 1:1 relationship between each record in EF_{GRDD} and the corresponding record in EF_{GSSID}.

Identifier: '6F0D'	Structure: linear fixed		Mandatory
Record size: 2 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Key record number	M	1
2	Group related data	M	1

See EF_{GRDS} for contents and coding.

10.3.14 EF_{GCK} (Group Cipher Keys)

This EF contains the group cipher keys associated with the group identities. There shall be a 1:1 relationship between each MGCK in EF_{MGCK} and the corresponding record of GCK in EF_{GCK}.

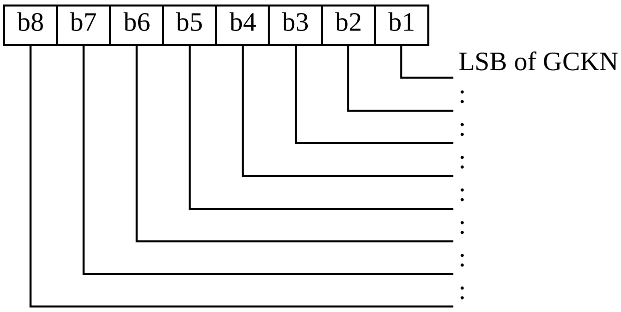
Identifier: '6F0E '	Structure: linear fixed		Optional
Record length: 12 bytes		Update activity: high	
Access Conditions: READ NEV (see note 1) UPDATE NEV (see note 2) INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1-2	GCKN	M	2
3-12	GCK	M	10
NOTE 1: There is no access to this EF over the SIM-ME interface. NOTE 2: GCK and GCKN are updated on the SIM by use of the TA41TA82 algorithm. NOTE 3: A record is free if no (static or dynamic) GSSI points to it.			

GCKN:

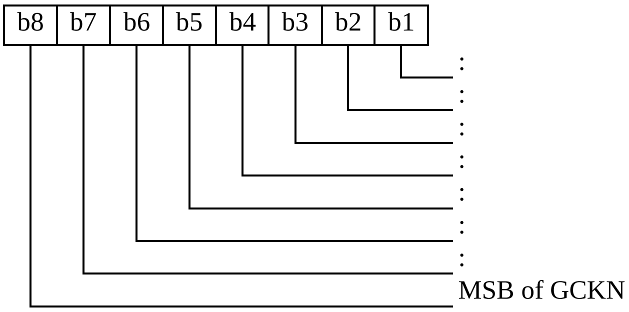
Contents: The Group Cipher Key Number is the identifier for a GCK used to associate it to one or more groups.

Coding:

Byte 1:



Byte 2:

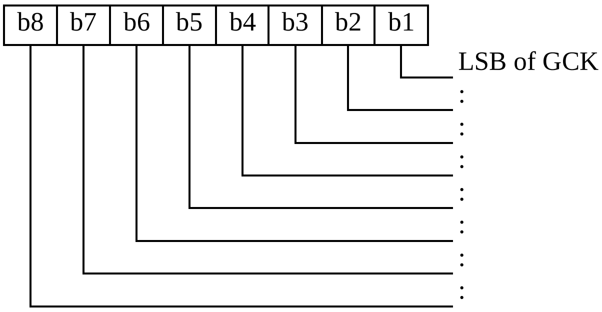


GCK:

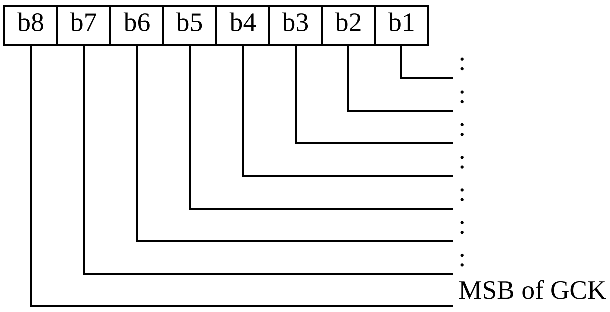
Contents: The Group Cipher Keys.

Coding: The key is stored in 10 bytes according to the following diagram:

Byte 1:



Byte 10:



10.3.15 EF_{MGCK} (Modified Group Cipher Keys)

This EF contains the modified group cipher keys associated with the group identities. There shall be a 1:1 relationship between each MGCK in EF_{MGCK} and the corresponding record of GCK in EF_{GCK}.

Identifier: '6F0F '	Structure: key	Optional
Record length: 12 bytes	Update activity: high	

Access Conditions:

READ	CHV1 (see note 1)
UPDATE	NEV (see note 2)
INVALIDATE	NEV
REHABILITATE	NEV

Bytes	Description	M/O	Length
1-2	GCK-VN	M	2
3-12	MGCK	M	10

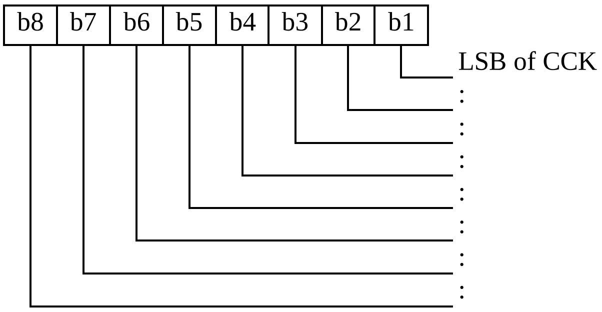
NOTE 1: Read access to this file is only possible by use of Read Key command.
NOTE 2: Updating of this EF is performed by the TA71 algorithm on the SIM.
NOTE 3: A record is free if no (static or dynamic) GSSI points to it.

GCK-VN:

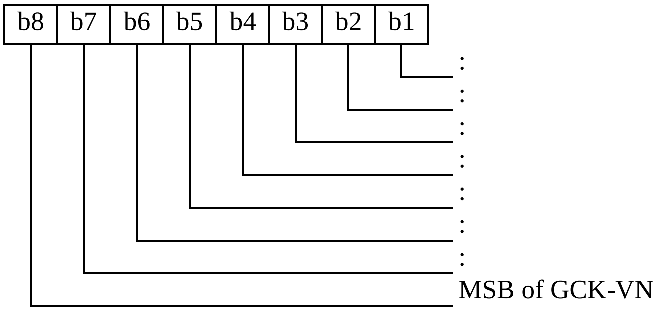
Contents: Group Cipher key Version Number

Coding:

Byte 1:



Byte 2:



MGCK:

Contents: The Modified Group Cipher Key.

Coding: See EF_{GCK}

10.3.16 EF$_{GINFO}$ (User's group information)

This EF contains the user's last active group, user's preferred group and information about using these group addresses.

Identifier: '6F10 '	Structure: transparent	Mandatory
File size: 9 bytes	Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

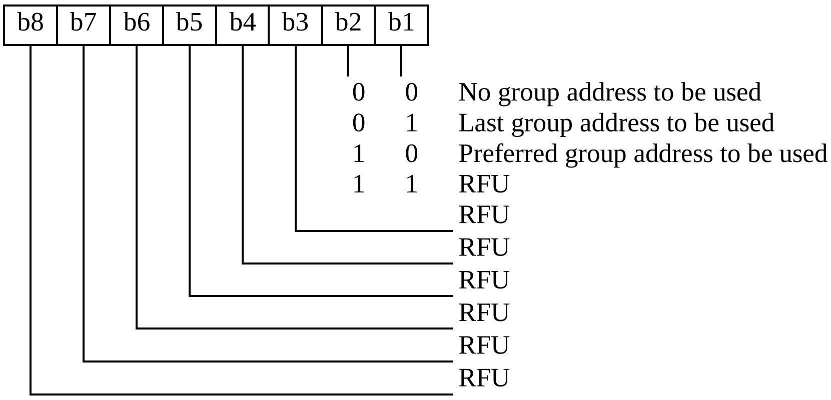
Bytes	Description	M/O	Length
1	Usage information	M	1
2	Network address record number of last active group	M	1
3-5	GSSI of the last active group	M	3
6	Network address record number of user's preferred group	M	1
7-9	GSSI of the user's preferred group	M	3

Usage information:

Contents: Two bits indicate the use of addresses.

Coding:

Byte 1:


b8 b7 b6 b5 b4 b3 b2 b1
0 0 No group address to be used
0 1 Last group address to be used
1 0 Preferred group address to be used
1 1 RFU
RFU
RFU
RFU
RFU
RFU
RFU

Network address record number of last active group:

Contents: Record number of the corresponding network address in EF$_{NWT}$.

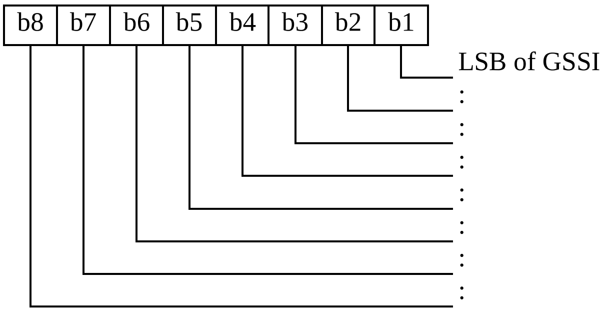
Coding: Binary. NULL value ('00') indicates that no GSSI is stored.

GSSI of the last active group:

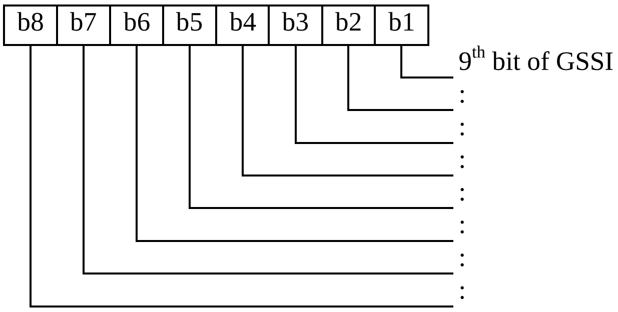
Contents: The short subscriber identity for the group that was last active.

Coding: Length of the GSSI is 24 bits.

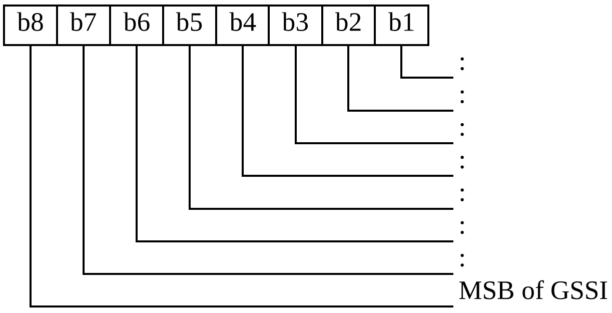
Byte 3:



Byte 4:



Byte 5:



Network address record number of user's preferred group:

Contents: Record number of the corresponding network address in EF_{NWT}.

Coding: binary. NULL value ('00') indicates that no GSSI is stored.

GSSI of the user's preferred group:

Contents: The short subscriber identity for the user's preferred group.

Coding: Length of the GSSI is 24 bits. Coded as GSSI of the last active group above, except with bytes 7-9.

NOTE: This record is updated at the beginning of a group call.

10.3.17 EF_{SEC} (Security settings)

This EF indicates the values for the security settings.

Identifier: '6F11'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

Access Conditions:	
READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Security settings	M	1

Security settings:

Contents: indicates whether the SIM requests a mutual authentication when it is authenticated by the SwMI, or whether the SIM requests authentication and the security class

Coding:

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1	
							0	Mutual authentication not required.
							1	Mutual authentication required;
						0		Authentication not required
						1		Authentication required;
				0	0			Security Class 1
				0	1			Security Class 2
				1	0			Security Class 3
				1	1			RFU
								RFU

10.3.18 EF_{FORBID} (Forbidden networks)

This EF contains the coding for Forbidden networks. It is read by the ME as part of the SIM initialization procedure and indicates networks which the MS shall not automatically attempt to access.

A network address is written to the EF if a network rejects a Location Update with the following causes "Illegal MS" and "Migration not supported" as in EN 300 392-2 [2]. The ME shall update the list by using the "next" mode of the update record command.

NOTE 1: By using the "next" mode in update operations the oldest record will be overwritten in the case the file is full.

NOTE 2: This EF should have at least as many records as is the expected amount of forbidden networks. Otherwise the ME may find the same forbidden networks in the beginning of every TETRA session and rewrite them to the list.

Identifier: '6F12 '	Structure: cyclic	Mandatory
Record length: 3 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

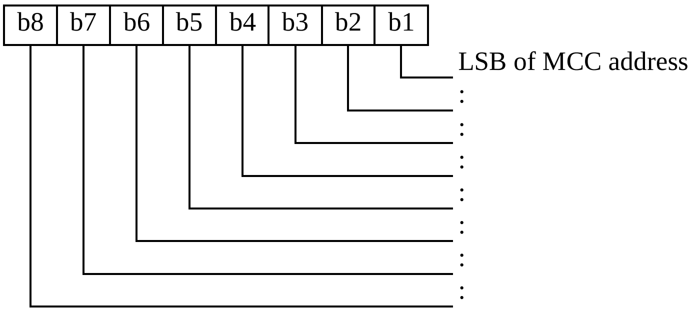
Bytes	Description	M/O	Length
1-3	Network address	M	3

Network address:

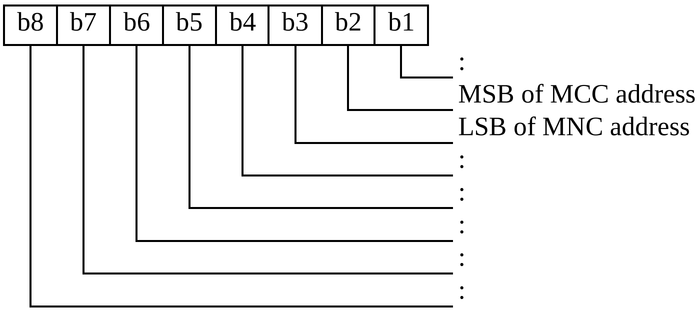
Contents: The address consists of MCC and MNC addresses, 10 and 14 bits respectively.

Coding: according to the following diagram. Empty records shall be set to 'FF'.

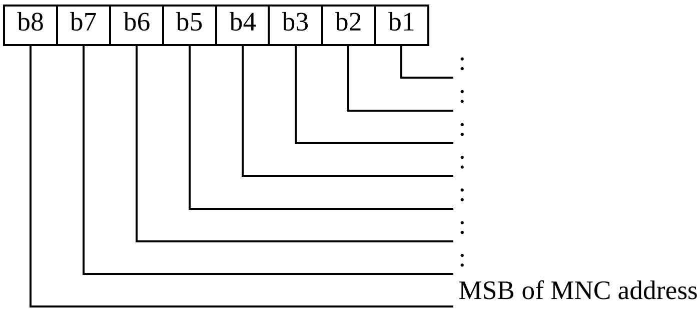
Byte 1:



Byte 2:



Byte 3:



10.3.19 EF$_{PREF}$ (Preferred networks)

This EF contains a list of preferred network addresses. The networks are listed in the order of preference. The first record corresponds to the highest preference.

Identifier: '6F13 '	Structure: linear fixed	Optional
Record length: 3 bytes	Update activity: low	

Access Conditions:
- READ CHV1
- UPDATE ADM
- INVALIDATE ADM
- REHABILITATE ADM

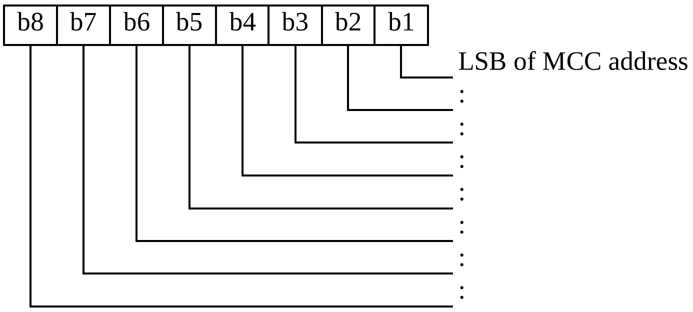
Bytes	Description	M/O	Length
1-3	Network address	M	3

Network address:

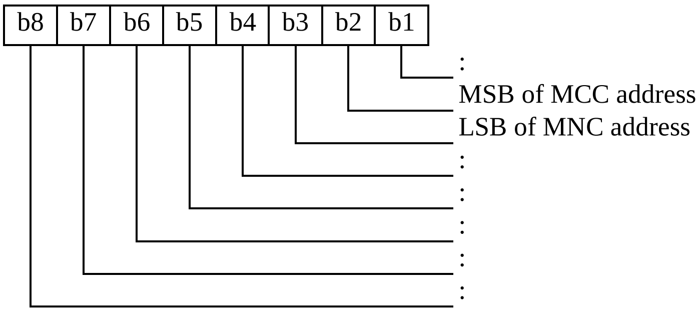
Contents: The address consists of MCC and MNC addresses, 10 and 14 bits respectively.

Coding: according to the following diagram. Empty records shall be set to 'FF'.

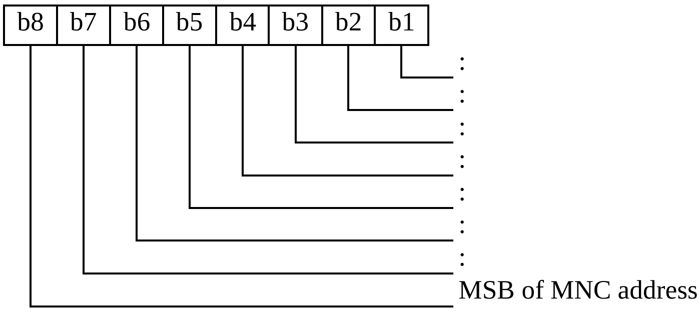
Byte 1:



Byte 2:



Byte 3:



10.3.20 EF$_{SPN}$ (Service Provider Name)

This EF contains the service provider name and appropriate requirements for the display by the ME.

Identifier: '6F14 '	Structure: transparent	Optional
File size: 17 bytes	Update activity: low	

Access Conditions:

READ	ALW
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

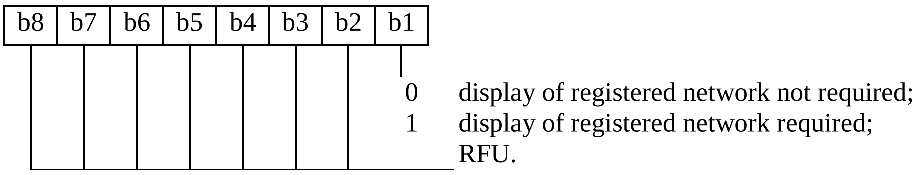
Bytes	Description	M/O	Length
1	Display Condition	M	1
2-17	Service Provider Name	M	16

Display condition:

Contents: Display condition for the service provider name in respect to the network.

Coding:

Byte 1:


b8
b7
b6
b5
b4
b3
b2
b1
0 display of registered network not required;
1 display of registered network required;
RFU.

Service provider name:

Contents: Service provider string to be displayed.

Coding: The string shall use the default 8-bit alphabet ISO 8859-1 [24]. The string shall be left justified. Unused bytes shall be set to 'FF'.

10.3.21 EF$_{LOCI}$ (Location information)

NOTE: This is an obsolete TETRA function which was specified in ETS 300 812 Edition 1 [8]. The function shall not be used by an ME supporting the present document or it's later versions. This file will be removed from later versions of the present document.

This EF contains the following information:

Alias Short Subscriber Identity (ASSI) or Visitor ASSI (V-ASSI);

Network address record number for ASSI or V-ASSI;

Location Area (LA);

Network address record number for Location Area.

Identifier: '6F15 '	Structure: transparent		Optional
File size: 7 bytes		Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			

Bytes	Description	M/O	Length
1 - 3	ASSI or V-ASSI	M	3
4	ASSI or V-ASSI Network address record number	M	1
5-6	Location Area	O	2
7	Location Area Network address record number	O	1

ASSI or V-ASSI:

Contents: Alias Short Subscriber Identity.

Coding: Short Subscriber Identity (SSI) according to EF_{ITSI}.

ASSI or V-ASSI Network address record number:

Contents: Network address record number for ASSI or V-ASSI. Refers to addresses in EF_{NWT}.

Coding: binary. NULL value ('00') indicates that no ASSI or V-ASSI stored.

Location Area:

Contents: Location Area of last successful registration.

Coding: As per EN 300 392-2 [2] (14 bits) with b8 and b7 of first byte RFU.

LA Network address record number:

Contents: Network address record number for LA. Refers to addresses in EF_{NWT}.

Coding: binary. NULL value ('00') indicates that no Location Area stored.

10.3.22 EF_{DNWRK} (Broadcast network information)

This EF contains information concerning the D-NWRK-BROADCAST according to EN 300 392-2 [2] It shall contain 32 records (see EN 300 392-2 [2]).

Storage of neighbour cell information may reduce the extent of a MSs search for MCCH carriers when selecting a cell.

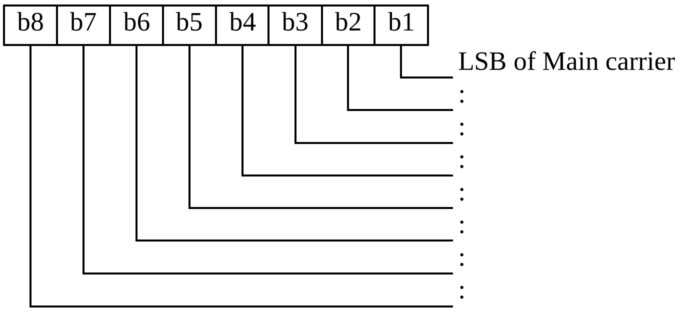
Identifier: '6F16 '	Structure: linear fixed		Mandatory
Record size: 3 bytes		Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			

Bytes	Description	M/O	Length
1 - 3	MCCH information	M	3 bytes

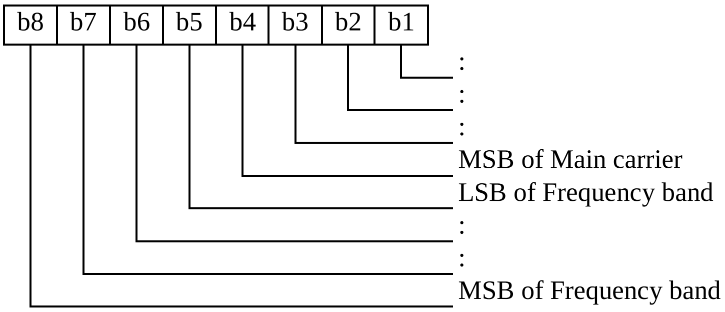
MCCH information:

Coding: The information is coded as defined in EN 300 392-2 [2] Free records are indicated in bit 7 of byte 3.

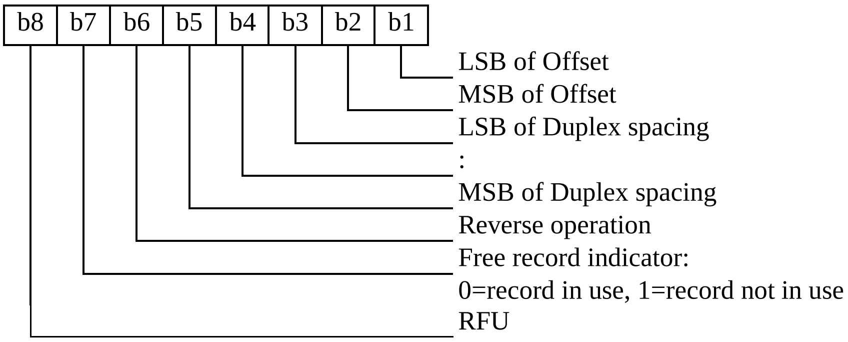
Byte 1:



Byte 2:



Byte 3:



10.3.23 EF$_{NWT}$ (Network table)

This EF contains the network part of the TETRA addresses. These addresses are used and updated by several EFs (EF$_{GSSIS}$, EF$_{GSSID}$, EF$_{GINFO}$, EF$_{GWT}$, EF$_{ADNTETRA}$, EF$_{SDNTETRA}$, EF$_{FDNTETRA}$, and EF$_{LNDTETRA}$). The records in these files make reference to particular network address records in this file using the record number of the network address.

Identifier: '6F17 '	Structure: linear fixed	Mandatory
Record size: 5 bytes	Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

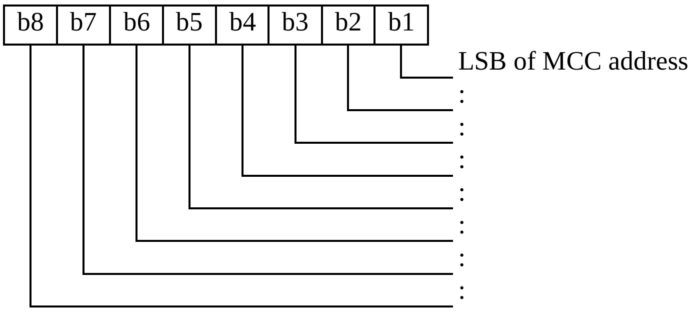
Bytes	Description	M/O	Length
1-3	Network address (MCC and MNC)	M	3
4—5	Record pointer counter	M	2

Network address:

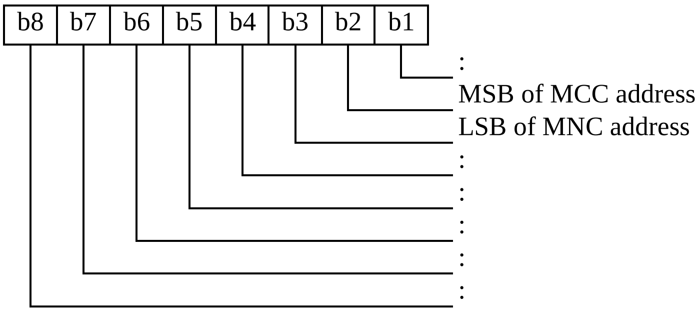
Contents: The address consists of MCC and MNC addresses, 10 and 14 bits respectively. The user's home address (from ITSI) is stored as the first record of the file.

Coding:

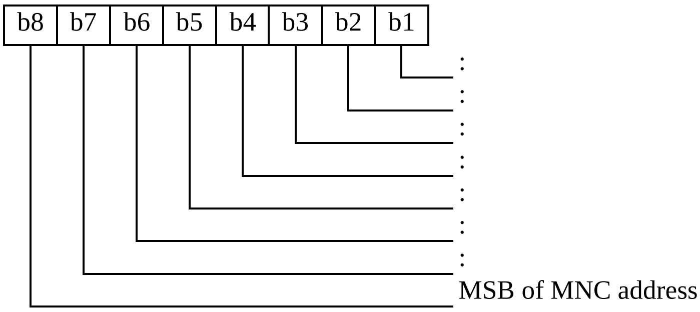
Byte 1:



Byte 2:



Byte 3:



Record pointer counter:

Contents: The records in this file can be referenced from several other files. This counter is incremented each time a new reference to a record is created. Also when the reference is deleted, this counter should be decremented.

Coding: Binary. NULL value ('00') indicates a free record.

NOTE: This file is updated by the ME when updating EFs which reference this file.

10.3.24 EF_{GWT} (Gateway table)

This EF contains the names and addresses for gateways in a TETRA network e.g. Private Automatic Branch Exchange (PABX). and Public Switched Telephone Network (PSTN). This file is referenced by EF_{ADNGWT}, EF_{FDNGWT}, EF_{LNDGWT}, EF_{SDNGWT}, EF_{ADN}, EF_{FDN}, EF_{LND} and EF_{SDN}. The files reference to this file using the record number of gateway names and addresses on this file.

NOTE: This implementation requires that there is one universally acknowledged TETRA address for PSTN gateways in all different networks.

Identifier: '6F18 '	Structure: linear fixed	Optional
Record size: 13 bytes	Update activity: high	

Access Conditions:

READ	CHV1
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1-8	Name	M	8
9	Network address record number	M	1
10-12	SSI of the gateway	M	3
13	Type	M	1
14	RFU	M	1

The name and address of the PSTN gateway is stored as the first record of the file

Name:

Contents: The alphanumeric name for the corresponding gateway.

Coding: The string shall use the default 8-bit alphabet. The string shall be left justified. Unused bytes shall be set to 'FF'.

Network address record number:

Contents: Record number of the corresponding network address in EF_{NWT}.

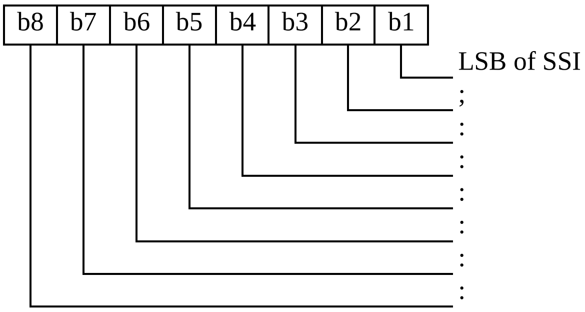
Coding: binary

SSI of the Gateway:

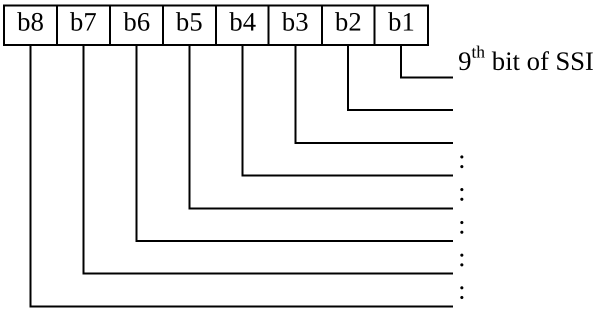
Contents: The short subscriber identity of the gateway used.

Coding: Length of the SSI is 24 bits.

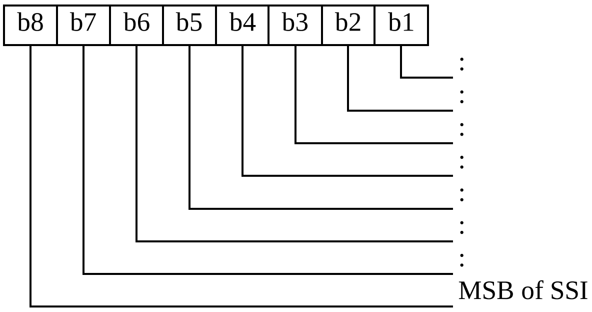
Byte 11:



Byte 12:



Byte 13:

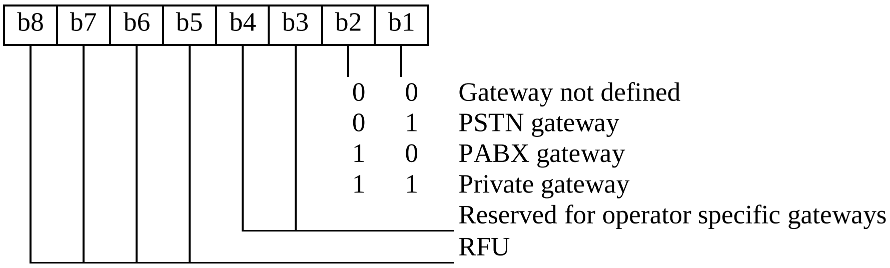


Type:

Contents: The type of gateway.

Coding:

Byte 1:



10.3.25 EF$_{CMT}$ (Call Modifier Table)

This EF indicates the values for the call modifiers required by the ME on a per call basis. These are intended to provide a sensible set of call modifiers for use where the user does not, or can not, enter them during call set-up. It is proposed that there are different sets of modifiers for different types of calls and that these sets are selected by the ME according to the call type. Alternatively, the ME may allow the user to select a set of call modifiers via the MMI. The alphanumeric field is intended to assist the user in selecting a proper call modifier.

To allow default values to be defined on subscription for each of the call types, the first 12 entries in the table are designated for particular call types in fixed positions. The user may add more call modifiers after the first 12 entries.

Each record in phonebooks may refer to a call modifier in this EF.

<table>
<tr><td colspan="2">Identifier: '6F19 '</td><td colspan="3">Structure: linear fixed</td><td>Optional</td></tr>
<tr><td colspan="3">Record length: X + 4 bytes</td><td colspan="3">Update activity: low</td></tr>
<tr><td colspan="6">Access Conditions:
READ CHV1
UPDATE CHV1/CHV2 (see note)
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="3">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1 to X</td><td colspan="3">Name</td><td>M</td><td>X</td></tr>
<tr><td>X + 1 to X + 4</td><td colspan="3">Call modifiers</td><td>M</td><td>4</td></tr>
<tr><td colspan="6">NOTE: Card issuer will choose between CHV1 or CHV2 protection.</td></tr>
</table>

Name:

Contents: An alphanumeric identifier for the set of call modifier values.

Coding: According to the default 8-bit alphabet ISO 8859-1 [24] A free record is indicated by filling this field with 'FF'.

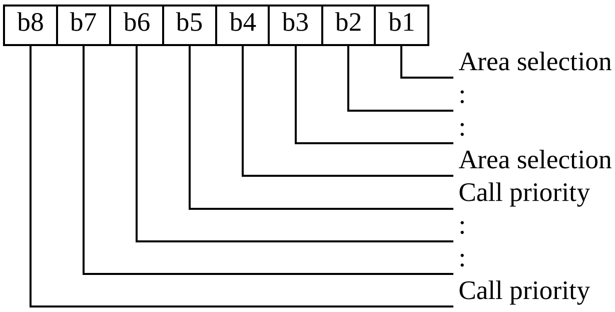
Call modifiers:

Contents: The file consists of the following pieces of information:

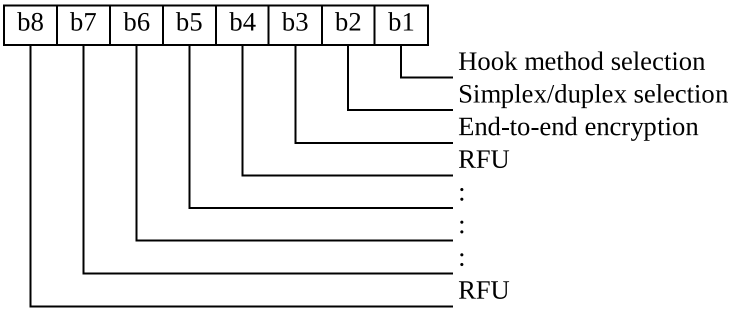
Area selection	4 bits;
Call priority	4 bits;
Hook method selection	1 bit;
Simplex/duplex selection	1 bit;
End-to-end encryption	1 bit;
Basic service information	16 bits.

Coding: All bits are coded into four bytes.

Byte 1:



Byte 2:



Bytes 3 and 4 are coded as "basic service information" in EN 300 392-2 [2]

Fixed call modifier sets:

the default call modifier sets are placed in EF_{CMT} in a standard order to allow selection of the set by call type.

Record in EF_{CMT}	**Call Type**	**Call features**
Record 1	Voice call	Intra-TETRA, individual call
Record 2	Voice call	Intra-TETRA, group call
Record 3	Voice call	Intra-TETRA, acknowledged group
Record 4	Voice call	Intra-TETRA, broadcast call
Record 5	Voice call	PABX call
Record 6	Voice call	PSTN call
Record 7	Circuit mode data call	Intra-TETRA, individual call
Record 8	Circuit mode data call	Intra-TETRA, group call
Record 9	Circuit mode data call	Intra-TETRA, acknowledged group
Record 10	Circuit mode data call	Intra-TETRA, broadcast call
Record 11	Circuit mode data call	PABX call
Record 12	Circuit mode data call	PSTN call

NOTE: This EF references the EN 300 392-2 [2]

10.3.26 EF_{ADNGWT} (Abbreviated Dialling Number with Gateways)

This EF contains ADNs. In addition it contains record numbers of the associated gateway, call modifier and gateway extension records.

NOTE: When calling to phone numbers contained in this EF from within a TETRA network, the gateway address is sent with the dialled number.

<table>
<tr><td colspan="2">Identifier: '6F1A '</td><td>Structure: linear fixed</td><td>Optional</td></tr>
<tr><td colspan="2">Record length: X+12 bytes</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="4">Access Conditions:
READ CHV1
UPDATE CHV1
INVALIDATE CHV2
REHABILITATE CHV2</td></tr>
<tr><td>Bytes</td><td>Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1 to X</td><td>Name</td><td>O</td><td>X</td></tr>
<tr><td>X+1</td><td>Length of number contents</td><td>M</td><td>1</td></tr>
<tr><td>X+2 to X+9</td><td>Dialling number</td><td>M</td><td>8</td></tr>
<tr><td>X+10</td><td>Gateway address record number</td><td>M</td><td>1</td></tr>
<tr><td>X+11</td><td>Call modifier record number</td><td>M</td><td>1</td></tr>
<tr><td>X+12</td><td>Gateway Extension1 record number</td><td>M</td><td>1</td></tr>
</table>

Name:

Contents: The alphanumeric name the user has assigned for corresponding dialling number.

Coding: According to the default 8-bit alphabet ISO 8859-1 [24].

Length of number contents:

Contents: this field gives the number of digits of the following " number" -field containing an actual BCD number. This means that the maximum value is 16, even when the actual ADN length is greater than 16 digits. When an ADN requires more than 16 digits it is indicated by the Gateway Extension 1 record number being unequal to 'FF'. The remainder is stored in the $\text{EF}_{\text{GWTEXT1}}$ with the remaining length of the overflow data being coded in the appropriate overflow record itself (see clause 10.3.27).

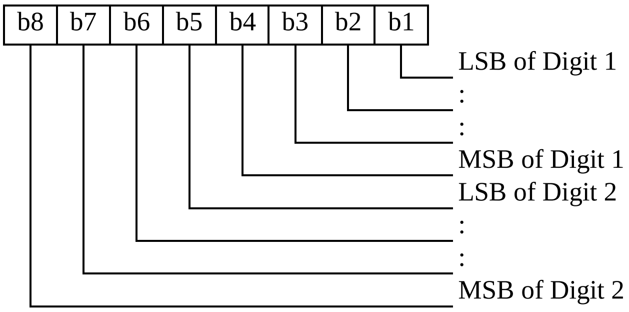
Coding: binary. NULL ('00') value indicates a free record.

Dialling number:

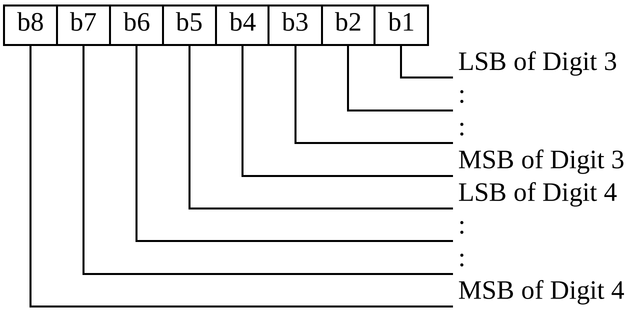
Contents: up to 16 digits of the number.

Coding: according to EN 300 392-2 [2]. If the dialling number is longer than 16 digits, the first 16 digits are stored in this data item and the overflow data is stored in an associated record in the $\text{EF}_{\text{GWTEXT1}}$. The record is identified by the Gateway Extension 1 record number. If ADN requires less than 16 digits, excess nibbles at the end of the data item shall be ignored.

Byte X+2



Byte X+3:



etc.

Gateway address record number:

Contents: This byte identifies the number of a record in the EF_{GWT} containing an associated gateway address. The use of this byte is optional. If it is not used it shall be set to 'FF'.

Coding: binary.

Call modifier record number:

Contents: This byte identifies the number of a record in the EF_{CMT} containing an associated call modifier information. The use of this byte is optional. If it is not used it shall be set to 'FF'.

Coding: binary.

Gateway Extension 1 record number:

Contents: This byte identifies the number of a record in the $EF_{GWTEXT1}$ containing an associated ADN overflow. The use of this byte is optional. If it is not used it shall be set to 'FF'.

Coding: binary.

10.3.27 $EF_{GWTEXT1}$ (Gateway Extension1)

This EF contains extension data of an ADNGWT or Last Number Dialled with gateway (LNDGWT). Extension data is caused by an ADNGWT or LNDGWT which is greater than the 16 digit capacity of the ADNGWT or LNDGWT EF. The remainder is stored in this EF as a record, which is identified by a specified identification byte inside the ADNGWT or LNDGWT EF.

Identifier: '6F1B '	Structure: linear fixed		Optional
Record length: 13 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Record Type	M	1
2 to 12	Extension data	M	11
13	Identifier	M	1

For contents and coding as defined in GSM 11.11 [9]

10.3.28 $EF_{ADNTETRA}$ (Abbreviated dialling numbers for TETRA network)

EF contains the phone numbers that are used when calling to a TETRA phone. The access strings for Supplementary services are stored in the same file.

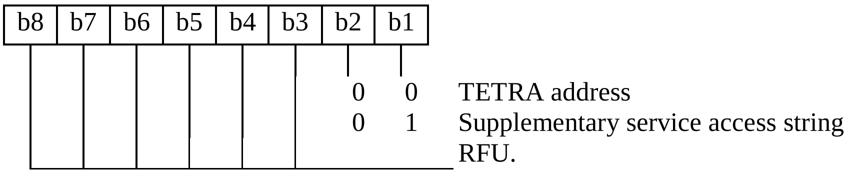
Identifier: '6F1C'	Structure: linear fixed		Optional
Record length: X + 7 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE CHV2 REHABILITATE CHV2			
Bytes	Description	M/O	Length
1	Type	M	1
2 to X+1	Name	M	X
X + 2	Network address record number	M	1
X + 3 to X + 5	TETRA address or Supplementary service access string	M	3
X + 6	Call modifier record number	M	1
X + 7	Extension A record number	M	1

Type:

Contents: One byte indicator to identify the entry type TETRA address or Supplementary service access string - field.

Coding:

Byte 1:



Name:

Contents: The alphanumeric name the user has assigned for corresponding phone number or Supplementary services access string.

Coding: According to the default 8-bit alphabet ISO 8859-1 [24].

Network address record number:

Contents: Record number of the corresponding network address. Network addresses are stored in EF_{NWT}.

Coding: Binary. NULL ('00') value indicates a free record. When storing the Supplementary service access strings to the TETRA address, this field is set to 'FF'.

Call modifier record number:

Contents: This byte identifies the number of a record in the EF_{CMT} containing an associated call modifier information. The use of this byte is optional. If it is not used it shall be set to 'FF'.

Coding: Binary.

TETRA address or Supplementary service access string:

Contents: The identity that is used when calling to a TETRA phone or Supplementary service strings to be stored.

Coding: When the field contains a TETRA address the field is binary-coded. When storing Supplementary service strings on this field, the digits and characters are BCD-coded according to EN 300 392-2 [2]

Extension A record number:

Contents: This byte identifies the number of a record in the EF_{EXTA} containing an associated supplementary services access string overflow. The use of this byte is optional. If it is not used, it shall be set to 'FF'.

Coding: Binary.

10.3.29 EF$_{EXTA}$ (Extension A)

This EF contains the overflow of a Supplementary service access string.

Identifier: '6F1D'	Structure: linear fixed	Optional
Record length: 20 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Length of extension data	M	1
2 to 19	Overflow data	M	18
20	Next record number	M	1

Length of extension data:

Contents: This field gives the number of digits of the following "Overflow data" -field containing an actual BCD number.

Coding: Binary. NULL ('00') value indicates a free record.

Overflow data:

Contents: Overflow data of a Supplementary services access string.

Coding: BCD according to EN 300 392-2 [2]

Next record number:

Contents: record number of the next extension record to enable storage of information longer than 18 bytes.

Coding: record number of next record. 'FF' identifies the end of the chain.

10.3.30 EF$_{FDNGWT}$ (Fixed dialling numbers with Gateways)

This EF contains FDN. In addition it contains record numbers of associated gateway, call modifier and gateway extension records.

NOTE 1: When calling to phone numbers contained in this EF from within a TETRA network, the gateway address is sent with the dialled number.

NOTE 2: Fixed dialling numbers are used for example in a situation when a supervisor in an organization fixes the numbers on a SIM card so that a worker of the organization may only call to work related numbers.

Identifier: '6F1E'	Structure: linear fixed	Optional
Record length: X + 12 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV2
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to X	Name	O	X
X + 1	Length of dialling number contents	M	1
X + 2 to X + 9	Dialling number	M	8
X + 10	Gateway address record number	M	1
X + 11	Call modifier record number	M	1
X + 12	Gateway Extension2 record number	M	1

For contents and coding of all data items see the respective data items of the EF_{ADNGWT}, with the exception that gateway extension records are stored in the $EF_{GWTEXT2}$.

10.3.31 $EF_{GWTEXT2}$ (Gateway Extension2)

This EF contains gateway extension data of an FDN (see Gateway Extension2 record number in clause 10.3.30). Gateway Extension data is caused by an FDN which is greater than the 16 digit capacity of the EF_{FDNGWT}. The remainder is stored in this EF as a record, which is identified by a specified identification byte inside the EF_{FDNGWT}.

Identifier: '6F1F'	Structure: linear fixed		Optional
Record length: 13 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV2 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Record Type	M	1
2 to 12	Extension data	M	11
13	Identifier	M	1

For contents and coding as defined in GSM 11.11[9]

10.3.32 $EF_{FDNTETRA}$ (Fixed dialling numbers for TETRA network)

EF contains the Fixed Dialling Numbers (FDN) to be used within TETRA network.

Identifier: '6F20'	Structure: linear fixed		Optional
Record length: X + 7 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV2 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Type	M	1
2 to X+1	Name	M	X
X + 2	Network address record number	M	1
X + 3 to X + 5	SSI of TETRA address	M	3
X + 6	Call modifier record number	M	1
X + 7	Extension B record number	M	1

For contents and coding of all data items see the respective data items of the $EF_{ADNTETRA}$.

10.3.33 EF$_{\text{EXTB}}$ (Extension B)

This EF contains the overflow of a Supplementary service access string.

Identifier: '6F21'	Structure: linear fixed	Optional

Record length: 20 bytes	Update activity: low

Access Conditions:

READ	CHV1
UPDATE	CHV2
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Length of extension data	M	1
2 to 19	Overflow data	M	18
20	Next record number	M	1

For contents and coding of all data items see the respective data items of the EF$_{\text{EXTA}}$.

10.3.34 EF$_{\text{LNDGWT}}$ (Last number dialled with Gateways)

This EF contains the last numbers dialled (LND). In addition it contains record numbers of associated gateway, call modifier and gateway extension records.

NOTE: When calling to phone numbers contained in this EF from within a TETRA network, the gateway address is sent with the dialled number.

Identifier: '6F22'	Structure: cyclic	Optional

Record length: X + 12 bytes	Update activity: high

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to X	Name	O	X
X + 1	Length of dialling number contents	M	1
X + 2 to X + 9	Dialling number	M	8
X + 10	Gateway address record number	M	1
X + 11	Call modifier record number	M	1
X + 12	Gateway Extension1 record number	M	1

Contents and coding: see EF$_{\text{ADNGWT}}$.

10.3.35 EF$_{LNDTETRA}$ (Last numbers dialled for TETRA network)

EF contains the last numbers dialled to TETRA phones within TETRA network.

Identifier: '6F23'	Structure: cyclic	Optional
Record length: X + 7 bytes	Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Type	M	1
2 to X	Name	M	X
X + 2	Network address record number	M	1
X + 3 to X + 5	SSI of TETRA address or Supplementary service access string	M	3
X + 6	Call modifier record number	M	1
X + 7	Extension A record number	M	1

For contents and coding of all data items see the respective data items of the EF$_{ADNTETRA}$.

10.3.36 EF$_{SDNGWT}$ (Service Dialling Numbers with gateway)

This EF contains the special user-non-modifiable Service Dialling Numbers (SDN) that are used when calling to a phone outside the TETRA network. In addition it contains record numbers of associated gateway, call modifier and gateway extension records.

NOTE: When calling to telephones contained in this EF from within a TETRA network, the gateway address is sent with the dialled number.

Identifier: '6F24'	Structure: linear fixed	Optional
Record length: X + 12 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to X	Name	O	X
X + 1	Length of dialling number contents	M	1
X + 2 to X + 9	Dialling number	M	8
X + 10	Gateway address record number	M	1
X + 11	Call modifier record number	M	1
X + 12	Gateway Extension3 record number	M	1

For contents and coding of all data items see the respective data items of the EF$_{ADNGWT}$ (see clause 10.3.25), with the exception that gateway extension records are stored in the EF$_{GWTEXT3}$.

10.3.37 EF$_{GWTEXT3}$ (Gateway Extension3)

This EF contains gateway extension data of an SDN (see Extension3 record number in clause 10.3.36). Gateway Extension data is caused by an SDN which is greater than the 16 digit capacity of the EF$_{SDNGWT}$. The remainder is stored in this EF as a record, which is identified by a specified identification byte inside the EF$_{SDNGWT}$.

Identifier: '6F25'	Structure: linear fixed	Optional
Record length: 13 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Record Type	M	1
2 to 12	Extension data	M	11
13	Identifier	M	1

For contents and coding as defined in GSM 11.11[9]

10.3.38 EF$_{SDNTETRA}$ (Service Dialling Numbers for TETRA network)

EF contains the user-non-modifiable phone numbers that are used when calling to a TETRA phone.

Identifier: '6F26'	Structure: linear fixed	Optional
Record length: X + 6 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Type	M	1
2 to X+1	Name	M	X
X + 2	Network address record number	M	1
X + 3 to X + 5	SSI of TETRA address	M	3
X + 6	Call modifier record number	M	1

For contents and coding of all data items see the respective data items of the EF$_{ADNTETRA}$.

10.3.39 EF$_{STXT}$ (Status message texts)

This EF contains text strings to be displayed upon receipt of precoded status message.

Identifier: '6F27'	Structure: linear fixed	Optional
Record length: X + 2 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-2	Message value	M	2
3 to X + 2	Message text	M	X

Message value:

Contents: The message value identifies the actual message.

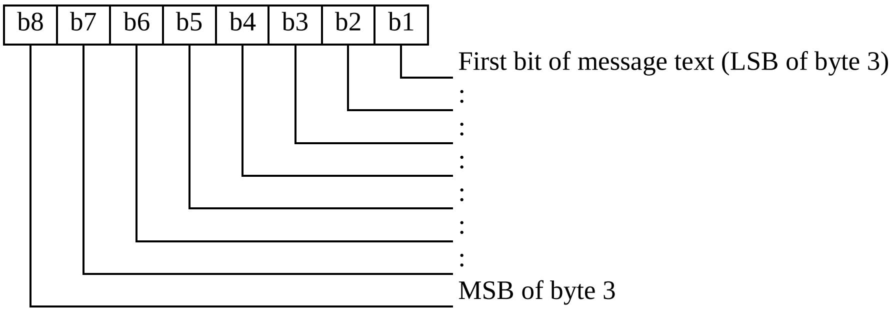
Coding: The message value is coded with two bytes as defined in EN 300 392-2 [2] A reserved ('0001'-'7FFF') value indicates an empty record.

Message text:

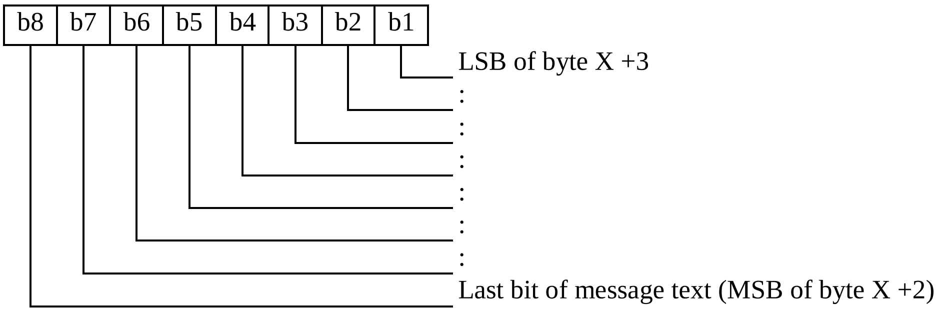
Contents: The message text contains the text string corresponding the message value and it is shown to the user instead of or with the message value.

Coding: The string shall use the default 8-bit alphabet ISO 8859-1 [24]. The message text is coded with X bytes. If the text is shorter than X bytes, the remaining bytes shall be filled with FF.

Byte 3:



Byte X+2:



NOTE: Of the precoded status messages only messages above and including the value of 32 768 are stored in this EF.

10.3.40 EF$_{MSGTXT}$ (SDS-1 message texts)

This EF contains text strings to be displayed upon receipt of an SDS-1 (user defined data 1) message.

Identifier: '6F28'	Structure: linear fixed		Optional
Record length: X + 2 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1-2	Message value	M	2
3 to X + 2	Message text	M	X

Message value:

Contents: The message value identifies the actual message.

Coding: The message value is coded with two bytes as defined in EN 300 392-2 [2]

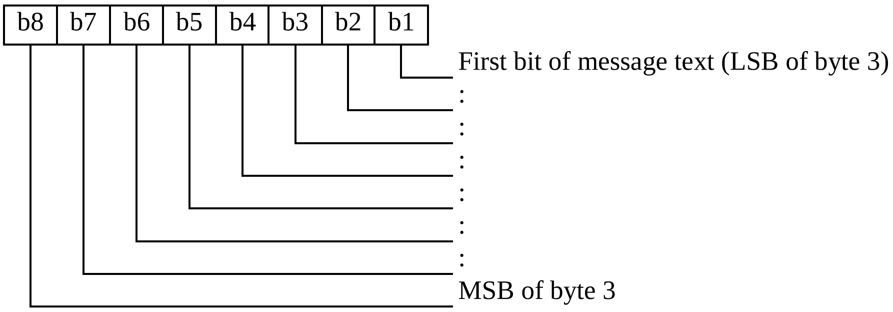
NOTE: User application knows which Message values are valid, because all values have been reserved for user application. Therefore the user application also knows which records contain valid data.

Message text:

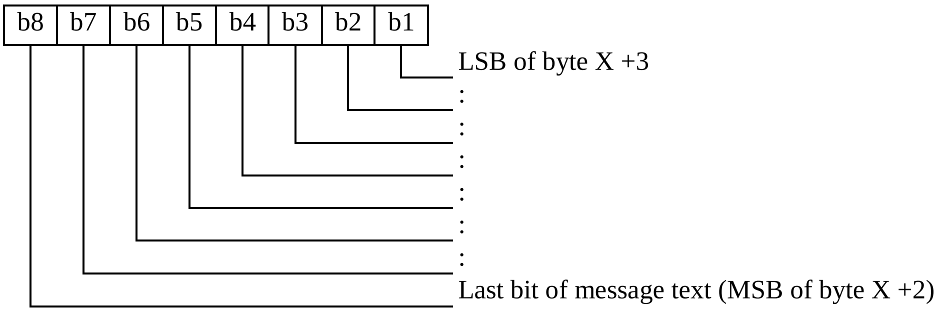
Contents: The message text contains the text string corresponding the message value and it is shown to the user instead of or with the message value.

Coding: The string shall use the default 8-bit alphabet ISO 8859-1 [24]. The message text is coded with X bytes. If the text is shorter than X bytes, the remaining bytes shall be filled with FF.

Byte 3:



Byte X+2:



NOTE: The SDS-1 messages are applicable to the user's home network only.

10.3.41 EF$_{SDS123}$ (Status and SDS type 1, 2 and 3 message storage)

This EF contains the numerical values of Status messages and SDS type 1, 2 or 3 messages (and associated parameters) which have either been received by the MS from the network, or are to be used as MS originated messages.

Identifier: '6F29'	Structure: linear fixed	Optional
Record length: 46 bytes	Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Message status	M	1
2-32	Message destination/source identifier	M	31
33 to 34	Message Index	M	2
35 to 37	Network Time	M	3
38-46	Message header and message (note 2)	M	9

Message status:

Contents: Status of the message stored.

Coding: byte 1.

b8	b7	b6	b5	b4	b3	b2	b1	
					0	0	0	Record not used
					0	1	0	RFU
					1	0	0	RFU
					1	1	0	RFU
					0	0	1	Message received by MS from Network; message read
					0	1	1	Message received by MS from network; message to be read
					1	0	1	MS originating message; message sent to the network
					1	1	1	MS originating message; message to be sent
								RFU

Message destination/source identifier:

For contents and coding see clause10.3.42

Message index:

Content:

It contains a message index. The Message Index will be incremented each time a new message is stored in this file. In case of an overflow the Message Index will be reset to 0.

Coding: 16 bits, binary

Network time:

Content: It indicates approximate reception time of the SDS message

Coding: 24 bits binary as defined in EN 300 392-2 [2]

Message header and message:

Contents: Contains information on transmitted or received messages.

Coding: The first byte is the short data type identifier as defined in EN 300 392-2[2].

Byte 38:

b8	b7	b6	b5	b4	b3	b2	b1	
						0	0	User Defined data 1
						0	1	User Defined data 2
						1	0	User Defined data 3
						1	1	User Defined data 4
								RFU

The other bytes are the user data 1,2,3 as defined in EN 300 392-2 [2].

10.3.42 EF_{SDS4} (SDS type 4 message storage)

This EF contains text strings (and associated parameters) which have either been received by the MS from the network, or are to be used as an MS originated message.

Identifier: '6F2A'	Structure: linear fixed	Optional
Record length: 255 bytes	Update activity: high	

Access Conditions:
READ CHV1
UPDATE CHV1
INVALIDATE ADM
REHABILITATE ADM

Bytes	Description	M/O	Length
1- 2	Message status	M	2
3-33	Message destination/source identifier(see note 1)	M	31
34	Protocol Identifier	M	1
35 to 35+ X-1	Message header (see note 2)	O	X
35+X to 36+X	Message Index	M	2
37+X to 39+X	Network Time	M	3
40+X to 41+X	Length Indicator	M	2
42+Xto 254	User Data	M	
255	Message extension record number	O	1

NOTE 1: The address length will be according to the address type (first byte in the message destination/source).
NOTE 2: For protocol identifier less than 128 there is no message header.

Message status:

Contents: It contains the status of the message stored and a pointer to the delivery report in case of originating message.

Coding:

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1	
					x	x	0	Free space
					0	0	1	Message received by MS from Network; message read
					0	1	1	Message received by MS from network; message to be read
					1	1	1	MS originating message; message to be sent
								RFU

Byte 2:

b8	b7	b6	b5	b4	b3	b2	b1	
			x	x	1	0	1	MS originating message; message sent to the network
			0	0	1	0	1	Status report not requested
			0	1	1	0	1	Status report requested but not yet received
			1	0	1	0	1	Status report requested, received but not stored in EF_{SDSR}
			1	1	1	0	1	Status report requested, received and stored in EF_{SDSR}
								RFU

Message destination/source identifier:

Contents: This data item contains:

For received message:

- The called party address

- Received address type

- Calling party address

For transmitted message:

- The called party address

- The calling/called address could be an SNA, SSI, TSI or external subscriber.

Coding:

The called party address:

Byte 3:

b8	b7	b6	b5	b4	b3	b2	b1	
					0	0	0	Short number address (SNA)
					0	0	1	Short subscriber identity (SSI)
					0	1	0	TETRA subscriber identity (TSI)
					0	1	1	External subscriber number
					1	0	0	RFU
					1	0	1	RFU
					1	1	0	RFU
					1	1	1	RFU
								RFU

Called party short number address:

Contents: the called party short number address consists of the SNA of the called user as defined in EN 300 392-2 [2]– byte 4:Address, bytes 5 to 17 set to “FF”

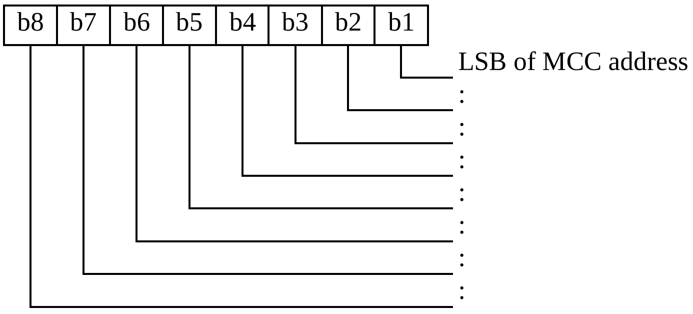
Called SSI

Contents: the SSI address of the called user as defined in EN 300 392-2[2]– byte 4 to 6:Address, byte 7 to 17 set to “FF”

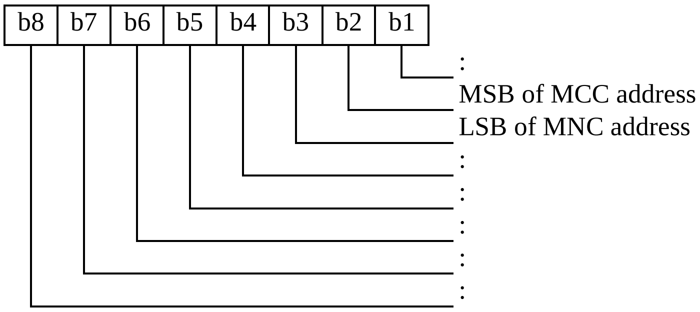
TETRA subscriber identity:

Contents: The TETRA subscriber identity as defined in ETS 300 392-1 [1], consists of Country Code (MCC), Network Code (MNC) and Short Subscriber Identity (SSI) – bytes 4 to 9: address, bytes 10 to 17 set to “FF”

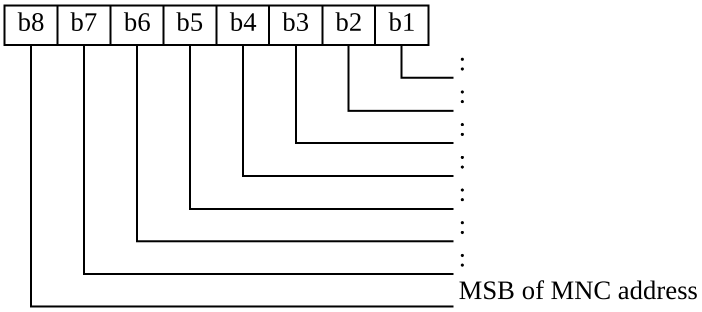
Byte 4:



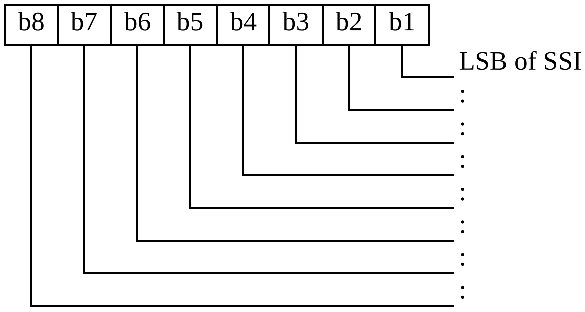
Byte 5:



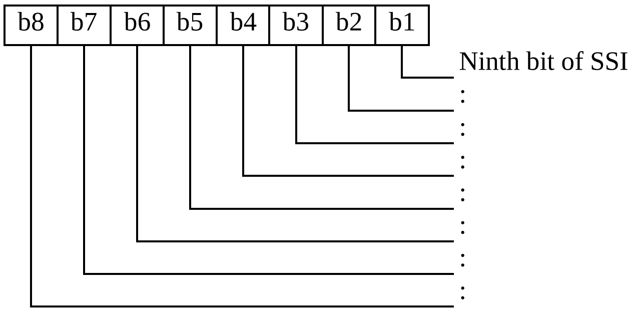
Byte 6:



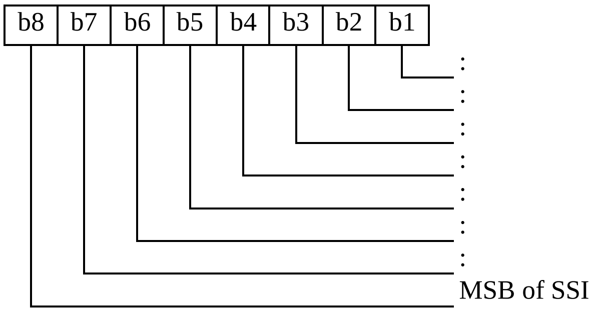
Byte 7:



Byte 8:



Byte 9:



External subscriber number:

Contents: It consists of the subscriber number and the gateway address record number.

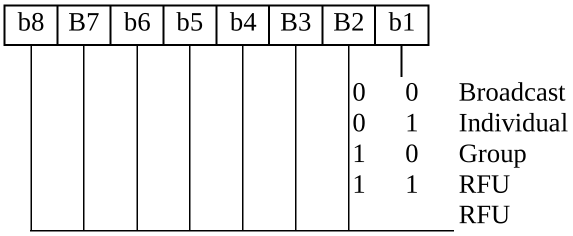
The gateway address record number identifies the number of a record in the EF_{GWT} containing an associated gateway address – byte 4: the record number

The subscriber number consists of the number of digits (less or equal to 24) and the digits. Each digit is as defined in EN 300 392-2[2]– byte 5: the number of digits, byte 6 to 6+n-1: the digits, the unused bytes set to “FF”.

Received address type:

Content: It consists of the address type of the called party.

Byte 18:



The calling party address

Same format as the called party address

Protocol Identifier:

Content: It shall indicate to the addressed entity application which type of application protocol is using the SDS service. See definition in EN 300 392-2 [2]

Coding: 1 byte as defined in EN 300 392-2 [2]

Message Header

Content:

For originating message it contains: the message reference, delivery report request, storage, validity period, service selection, forward address (only in case of storage).

For terminating message, it contains: the message reference, delivery report request, storage, validity period, short form report, and forward address.

Coding:

For originating message:

Message reference:

Each SDS-TL message carrying a SDS-TL data transfer service PDU shall contain a message reference. See definition in EN 300 392-2 [2]

Coding:

1 byte – “FF” – message to be sent, otherwise the message reference used in the message sent to the network.

Delivery report request:

2 bits as defined in EN 300 392-2 [2] (b1-b2 of byte 2 of message header)

Storage:

1 bit as defined in EN 300 392-2 [2] (b8 of byte2 of message header)

Validity Period:

5 bits as defined in EN 300 392-2 [2] (b1-b5 of byte 3 of message header)

Service Selection:

1 bit as defined in EN 300 392-2 [2] (b8 of byte 3 of message header)

Forward Address:

Same definition as the Message destination/source - only in case of storage.

For terminating message:

Message reference:

Each SDS-TL message carrying a SDS-TL data transfer service PDU shall contain a message reference. See definition in EN 300 392-2 [2]

Coding:

1 byte – “FF” – message to be sent, otherwise the message reference used in the message sent to the network.

Delivery report request:

2 bits as defined in EN 300 392-2 [2] (b1-b2 of byte 2 of message header)

Storage:

1 bit as defined in EN 300 392-2 [2] (b8 of byte 2 of message header)

Validity Period:

5 bits as defined in EN 300 392-2 [2] (b1-b5 of byte 3 of message header)

Short form report:

2 bits as defined in EN 300 392-2 [2] (b7-b8 of byte 3 of message header)

Forward Address:

Same definition as the Message destination/source - only in case of storage.

Message index:

Content:

It contains a message index. The Message Index will be incremented each time a new message is stored in this file. In case of an overflow the Message Index will be reset to 0

Coding:

16 bits, binary

Network time:

Content:

It indicates approximate reception time of the SDS message

Coding:

24 bits binary as defined in EN 300 392-2 [2]

Length Indicator:

Content:

It contains the length in bits of the user data

Coding:

11 bits, binary

User Data:

Content:

It contains the user data, as defined in EN 300 392-2 [2].

Byte 255 Message Extension record number:

Contents: This byte identifies the number of a record in the $\text{EF}_{\text{MSGEXT}}$ containing an associated message overflow. The use of this byte is optional. If it is not used, it shall be set to 'FF'.

Coding: Binary.

10.3.43 $\text{EF}_{\text{MSGEXT}}$ (Message Extension)

This EF contains the overflow of an SDS-4 message which is longer than the space reserved for it in EF_{SDS4}.

Identifier: '6F2B'	Structure: linear fixed		Optional
Record length: 16 bytes		Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1-16	Overflow message	M	16

Overflow message:

Contents: Overflow data of a SDS-4 message exceeding the length reserved for it in $EF_{SDS4.}$

Coding: As defined in EN 300 392-2 [2]. All bytes following the PDUs shall be filled with 'FF'.

NOTE: A free record is not pointed to by any record in EF_{SDS4}.

10.3.44 EF_{EADDR} (Emergency addresses)

The user (or the organization) can determine the address to which an emergency call is initiated; to a predetermined address or to the group last used by the user. The selection is controlled by the addresses stored in EF_{EADDR}.

Where a data call type is selected, the ESource field indicates the preferred source of the data to be included in the message for status, SDS-1, SDS-2, SDS-3 and SDS-4 messages. In each case the data content can be a pre-defined value stored in EF_{SDS123} or EF_{SDS4} (or a data field obtained from an application running in the terminal).

Identifier: '6F2C'	Structure: linear fixed		Mandatory
Record size: 17 bytes		Update activity: low	
Access Conditions: READ ALW UPDATE CHV1/CHV2 (see note) INVALIDATE ADM REHABILITATE ADM			
Bytes	**Description**	**M/O**	**Length**
1	Emergency call definition	M	1
2-17	Emergency address	M	16
NOTE: Card issuer will choose between CHV1 or CHV2 protection.			

Emergency call definition:

Contents: One byte indicating the call type and the emergency address type coded on the Emergency address field, and the source of the message content for status and data calls.

Coding:

b1-b4: Emergency call type

b5-b8: Call setup parameters

b5: Source of the data to be transmitted in the emergency data message

b6-b7: Emergency call type

b8: Simplex/Duplex

NOTE 1: An empty record is indicated by NULL ('F') value in bits b1-b4.

b8	b7	b6	b5	b4	b3	b2	b1	
				0	0	0	0	TETRA address
				0	0	0	1	DMO address
				0	0	1	0	PABX address (gateway and External subscriber number)
				0	0	1	1	PSTN number (gateway and External subscriber number)
				0	1	0	0	Last active group address
				0	1	0	1	RFU
				0	1	1	0	RFU
				0	1	1	1	RFU
				1	0	0	0	Status/SDS123 msg record number
				1	0	0	1	SDS4 message record number
				1	0	1	0	RFU
				1	0	1	1	RFU
				1	1	0	0	RFU
				1	1	0	1	RFU
				1	1	1	0	RFU
				1	1	1	1	Record contains no valid data
			0					Predefined and stored in EF_{EADDR}
			1					From an application in the terminal
	0	0						Point-to-Point
	0	1						Point to Multipoint
	1	0						Point-to-Multipoint acknowledged
	1	1						Broadcast
0								Simplex
1								Duplex

Emergency address:

Contents:

The address that can be used when the user initiates an emergency call. The type of call is determined by byte 1.

In the case of a TETRA address the emergency address consists of the ITSI (or GTSI) of the called party.

In the case of a DMO address the emergency address consists of the ITSI (or GTSI) of the called party and the DMO channel number.

In the case of a PABX address the emergency address consists of the PABX Gateway and the External Subscriber number. (see coding)

In the case of a PSTN address the emergency address consists of the PSTN Gateway and the external subscriber number. (see coding)

In the case of the last active group address, the address field in EF_{EADDR} is unused - the address for the emergency call should be obtained from EF_{GINFO}.

In the case of status, SDS-1, SDS-2, SDS-3 and SDS-4 messages the content of this data item consists of the message record number in SDS123 or SDS4 as appropriate.

Coding:

In the case of a TETRA address, according to EF_{ITSI}.

In the case of a DMO address, according to EF_{ITSI} followed by the 24 bit DMO channel number, coded according to EF_{DMOCh}.

In the case of a PABX number, the Gateway ITSI is coded according to EF_{ITSI} and the External Subscriber number is BCD coded as defined in EN 300 392-2 [2].

The structure will be as following:

Byte 2: Length of BCD number

Byte 3: Gateway address record number

Byte 4-16: Dialling Number

Byte 17: Gateway Extension 1 record number

In the case of a PSTN number, the Gateway ITSI is coded according to EF$_{ITSI}$ and the external PSTN address is BCD coded according to EN 300 392-2 [2]

The structure will be as following:

Byte 2: Length of BCD number

Byte 3: Gateway address record number

Byte 4-16: Dialling Number

Byte 17: Extension 1 record number

In the case of the last used group address, this field is unused - the address for the call to be obtained from EF$_{GINFO}$.

NOTE 2: The emergency addresses are stored in order of precedence.

10.3.45 EF$_{EINFO}$ (Emergency call information)

This EF contains information about setting up and continuing an emergency call.

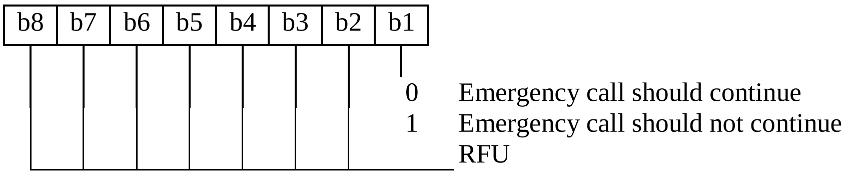
Identifier: '6F2D'	Structure: transparent		Mandatory
File size: 2 bytes		Update activity: low	
Access Conditions: READ ALW UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Emergency call continuation	M	1
2	Current emergency call record number	M	1

Emergency call continuation:

Contents: A flag indicating whether an interrupted emergency call should continue at power-on.

Coding:

Byte 1:



0 Emergency call should continue
1 Emergency call should not continue
RFU

Current emergency call record number:

Contents: One byte field available to the emergency application to store on the SIM information pertaining to an emergency call in progress, typically to cater for the possibility of unexpected power-down. It may be the record number of the record in EF$_{EADDR}$ used to set up the emergency call currently in progress. A zero value indicates that no call is in progress.

Coding: Binary.

10.3.46 EF$_{DMOCh}$ (DMO channel information)

This EF contains a selection of DMO channels. One or more of the channels may be designated as emergency channel(s) to be used for emergency calls within DMO operation.

NOTE: The information in the following EF may not be accurate with respect to ETS 300 396 series. This EF will be updated accordingly in Edition 3.

Identifier: '6F2E'	Structure: linear fixed	Optional
Record size: 4 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	DMO channel type	M	1
2-4	DMO channel number	M	3

DMO channel type:

Contents: This field contains the DMO channel type information.

Coding: The type is coded in the first bit of the first byte: emergency='01'; regular='00'. NULL ('FF') value indicates an empty record. All other values are reserved.

DMO channel number:

Contents: This field contains the DMO channel definition.

Coding: As defined in EN 300 392-2 [2].

10.3.47 EF$_{MSCh}$ (MS allocation of DMO channels)

This EF contains a bitmap which allocates a subset of the DMO channels in EF$_{DMOCh}$. There shall be one bit corresponding to each record in EF$_{DMOCh}$.

NOTE 1: The information in the following EF may not be accurate with respect to ETS 300 396 series. This EF will be updated accordingly in Edition 3.

Identifier: '6F2F '	Structure: transparent	Optional
File size: X bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

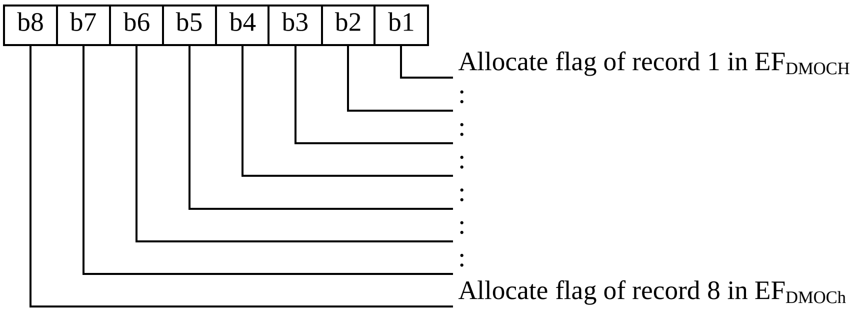
Bytes	Description	M/O	Length
1	Allocation flag 1 to 8	M	1
X	Allocation flag 8*X-7 to 8*X	M	1

NOTE 2: The value of X should be sufficiently large to accommodate all the records in EF_{DMOCh}.

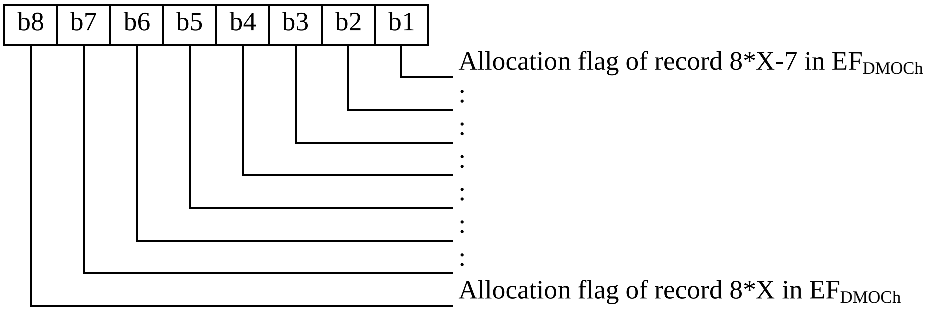
Allocation flag:

Coding: Channel is allocated=1, channel is not allocated=0.

Byte 1:


b8
b7
b6
b5
b4
b3
b2
b1
Allocate flag of record 1 in EF_{DMOCH}
:
:
:
:
:
:
Allocate flag of record 8 in EF_{DMOCh}

Byte X:


b8
b7
b6
b5
b4
b3
b2
b1
Allocation flag of record 8*X-7 in EF_{DMOCh}
:
:
:
:
:
:
Allocation flag of record 8*X in EF_{DMOCh}

10.3.48 EF_{KH} (List of Key Holders)

This EF contains a list of those ITSI numbers that can act as a key holder for this subscriber's ITSI.

NOTE: The information in the following EF may not be accurate with respect to ETS 300 396 series. This EF will be updated accordingly in Edition 3.

Identifier: '6F30 '	Structure: transparent		Optional
Record size: 6 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1 - 6	Key holder ITSI	M	6

Key holder ITSI;

Contents: Key holder ITSI consists of MCC, MNC and ISSI.

Coding: As in EF_{ITSI}. Record filled with NULL ('FF') value indicates no ITSI is stored.

10.3.49 $\text{EF}_{\text{REPGATE}}$ (DMO repeater and gateway list)

This EF contains a list of those DMO repeaters, gateways and REP/GATEs that this subscriber is allowed to use. Each address is 10 bits long. DMO equipment type is also identified.

NOTE: The information in the following EF may not be accurate with respect to ETS 300 396 series. This EF will be updated accordingly in Edition 3.

Identifier: '6F31'	Structure: linear fixed	Optional
Record size: 2 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

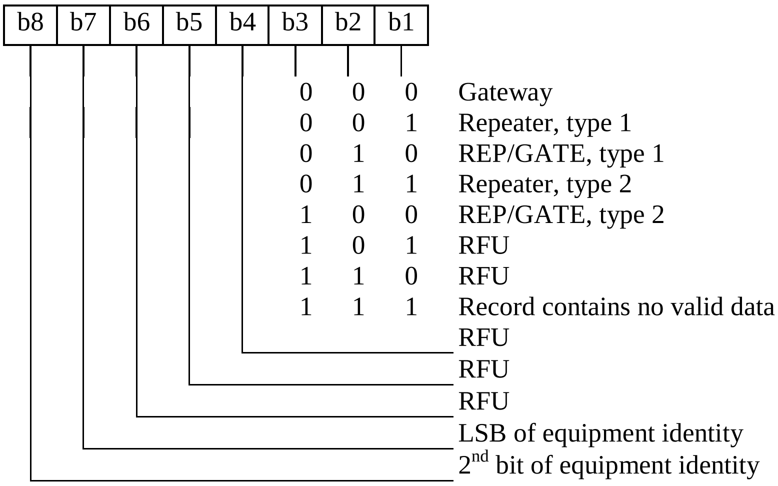
Bytes	Description	M/O	Length
1-2	DMO equipment type and identity	M	2

DMO equipment type and identity:

Contents: This field contains the DMO equipment type and the first part of its identity.

Coding:

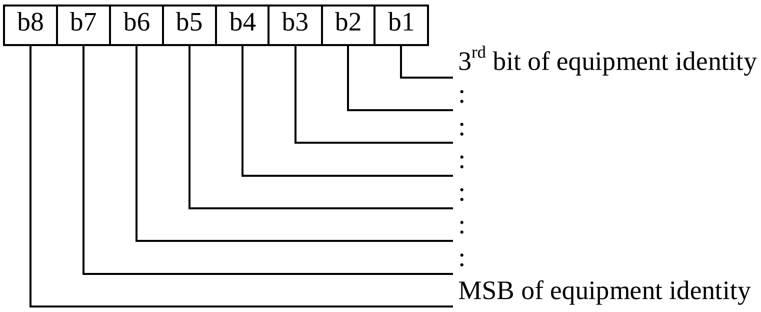
Byte 1:



b8	b7	b6	b5	b4	b3	b2	b1	
					0	0	0	Gateway
					0	0	1	Repeater, type 1
					0	1	0	REP/GATE, type 1
					0	1	1	Repeater, type 2
					1	0	0	REP/GATE, type 2
					1	0	1	RFU
					1	1	0	RFU
					1	1	1	Record contains no valid data

b4: RFU
b5: RFU
b6: RFU
b7: LSB of equipment identity
b8: 2nd bit of equipment identity

Byte 2:



b1: 3rd bit of equipment identity
b2: :
b3: :
b4: :
b5: :
b6: :
b7: :
b8: MSB of equipment identity

10.3.50 EF$_{AD}$ (Administrative data)

This EF contains information concerning the mode of operation according to the type of SIM, such as normal operation, type approval (to allow specific use of the ME during type approval procedures of e.g. the radio equipment) or others.

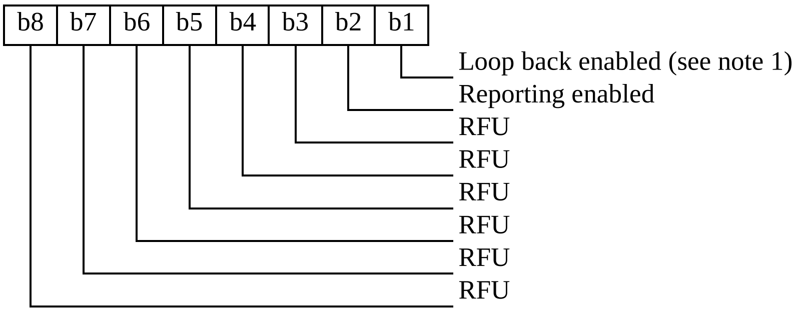
Identifier: '6F32'	Structure: transparent		Mandatory
File size: 1 byte		Update activity: low	
Access Conditions: READ ALW UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	MS operation mode	M	1 byte

MS operation mode:

Contents: mode of operation for the MS.

Coding:

Byte 1:



NOTE 1: Loop back enabled and security/authentication disabled (see ETS 300 394-2 [5]).

NOTE 2: The coding '00' means normal operation.

10.3.51 EF$_{PREF\_LA}$ (Preferred location areas)

This EF defines the preferred location area.

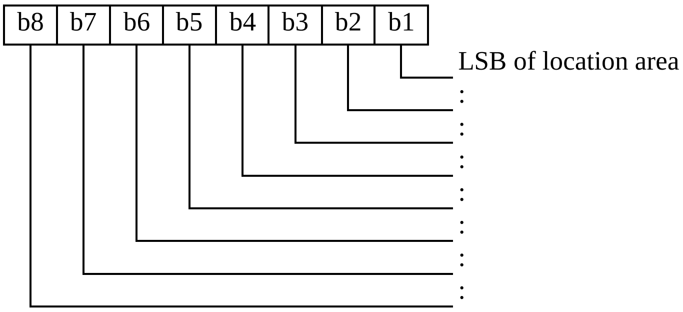
Identifier: '6F33'	Structure: transparent		Optional
File size: 2 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1-2	Preferred location area	M	2

Preferred location area:

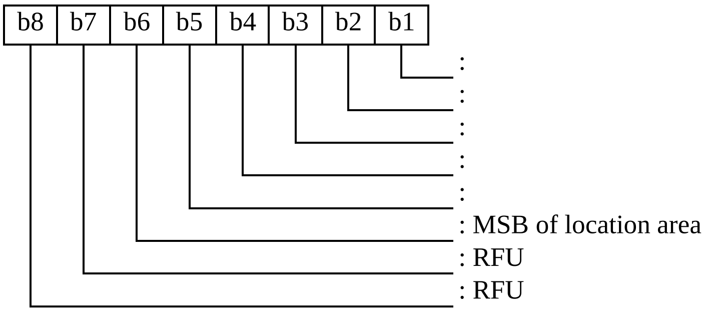
Contents: a list of preferred location areas.

Coding: Each element is coded in 2 bytes with the two highest order bits of the 2nd byte RFU. The first element (bytes 2 and 3) is shown below. See also EN 300 392-7 [3].

Byte 1:



Byte 2:



NOTE: This LA is intended to be used during cell re-selection, the procedures for which are outside the scope of the present document. See EN 300 392-2 [2].

10.3.52 EF$_{LNDComp}$ (Composite LND file)

This EF contains a pointer to the LND entries in EF$_{LND}$, EF$_{LNDGWT}$ and EF$_{LNDTETRA}$.

Identifier: '6F34'	Structure: cyclic		Optional
Record length: 3 bytes		Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1-2	Elementary File ID	M	2
3	Record No. in corresponding LND EF	M	1

Elementary File ID:

Contents: The ID of the file in which the LND record is stored.

Coding:

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1	
						0	0	EF within DF_{TETRA}
						0	1	EF within $DF_{TELECOM}$
						1	0	RFU
						1	1	RFU
								RFU

Byte 2:

b8	b7	b6	b5	b4	b3	b2	b1	
								2nd byte of file identifier

Record No. in corresponding LND Elementary File:

Contents: The record number of the LND.

Coding: Binary.

NOTE: This file shall be updated when any of the files EF_{LND}, EF_{LNDGWT} or $EF_{LNDTETRA}$ is updated.

10.3.53 $EF_{DFLTSTSTGT}$ (Status Default Target)

This EF contains information concerning the default target for status message texts.

Identifier: '6F35'	Structure: transparent		Optional
File size:16 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1	Acknowledgement required	M	1 byte
2	Address Type	M	1 byte
3-16	Address (see Note 1)	M	14 bytes
NOTE 1: The address length will be according to the address type. The unused bytes will be set to “FF”			

Acknowledgement required:

Contents: Indicates if an acknowledgement is required.

Coding:

Byte 1:

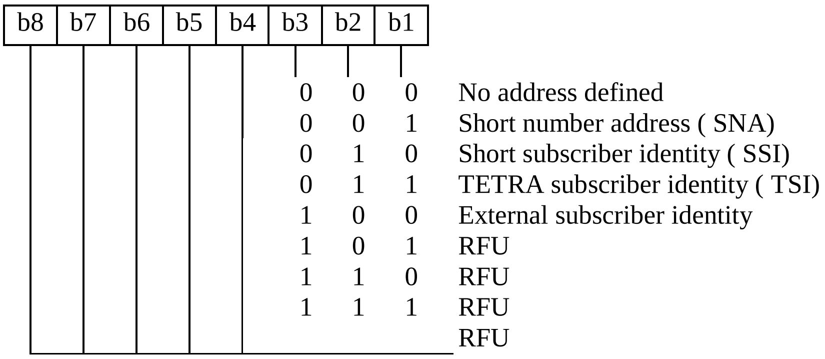
b8	b7	b6	b5	b4	b3	b2	b1	
x	x	x	x	x	x	x	0	acknowledgement required
x	x	x	x	x	x	x	1	no acknowledgement required

Address Type:

Contents: This data item contains the target address type.

Coding:

Byte 2:



b8	b7	b6	b5	b4	b3	b2	b1	
					0	0	0	No address defined
					0	0	1	Short number address (SNA)
					0	1	0	Short subscriber identity (SSI)
					0	1	1	TETRA subscriber identity (TSI)
					1	0	0	External subscriber identity
					1	0	1	RFU
					1	1	0	RFU
					1	1	1	RFU
								RFU

Address:

Contents: The address could be: a short number address, or an SSI, or a TETRA subscriber identity or an external subscriber identity:

Called party short number address:

Coding: the called party short number address consists of the SNA of the called user as defined in EN 300 392-2 [2] – byte 3 = Address, bytes 4-16 set to "FF"

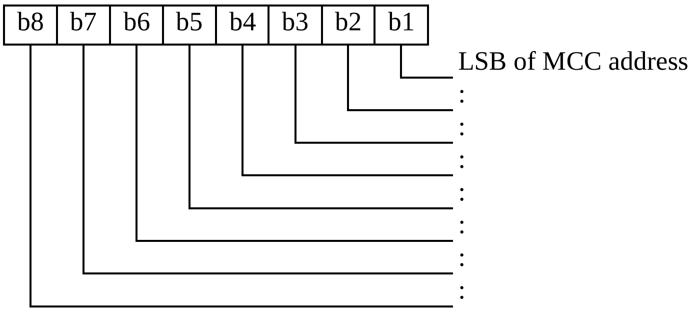
Called party SSI

Coding: the SSI address of the called user as defined in EN 300 392-2 [2] – bytes 3 to 5 = Address, byte 6-16 set to "FF"

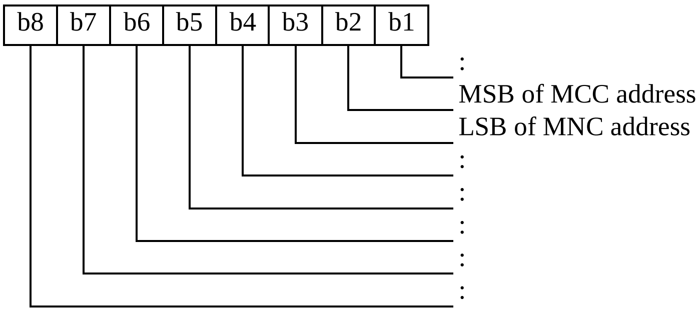
TETRA subscriber identity:

Coding: The TETRA subscriber identity as defined in ETS 300 392-1 [1], consists of Country Code (MCC), Network Code (MNC) and Short Subscriber Identity (SSI): byte 3-8 = address, bytes 9-16 set to "FF"

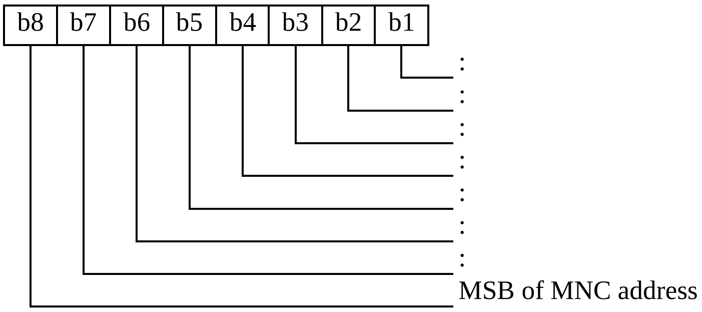
Byte 3:



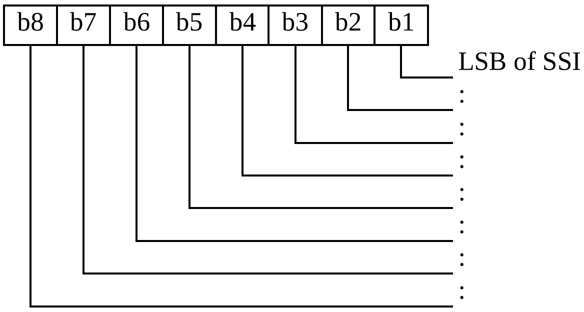
Byte 4:



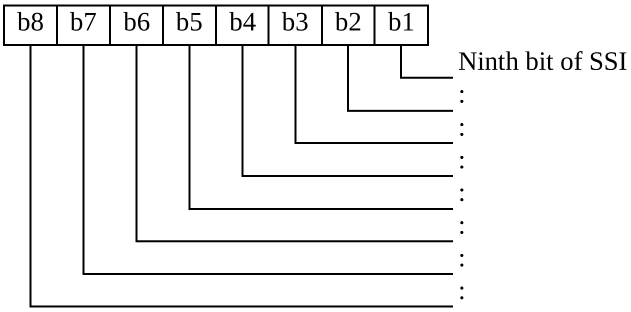
Byte 5:



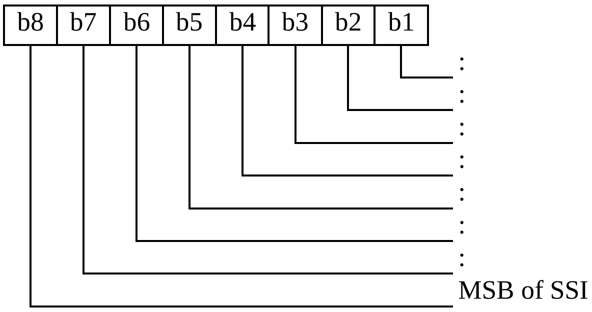
Byte 4:



Byte 5:



Byte 6:



External subscriber identity:

Contents: It consists of the external subscriber number and the gateway address record number.

The gateway address record number identifies the number of a record in the EF$_{GWT}$ containing an associated gateway address - byte 3 is the number of the record in the EF$_{GWT}$

The external subscriber number consists of the number of digits (less or equal to 24) and the digits. Each digit is as defined in EN 300 392-2 [2] – byte 4 – the number of digits, byte 5 to 5+n-1 the digits, all unused set to “FF”.

10.3.54 EF$_{SDSMEM\_STATUS}$ (SDS Memory Status)

This EF contains storage information relating to the SDS service.

The provision of this EF is associated with EF$_{SDS123}$ and/or EF$_{SDS4}$. The files shall be present together, or both absent from the SIM.

Identifier: '6F36'	Structure: transparent	Optional
File length: 7 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Last used TP-Message Reference	M	1 bytes
2	SDS4 “Memory capacity exceeded” notification flag	M	1 bytes
3	SDS123 memory capacity exceeded notification flag	M	1 bytes
4 to 5	SDS4 last used message index	M	2 bytes
6 to 7	SDS123 last used message index	M	2 bytes

Last used Transport Protocol (TP)-Message Reference

Contents:

The value of the TP-Message Reference parameter in the last mobile originated short message, as defined in ETS 300 392–2[2].

Coding:

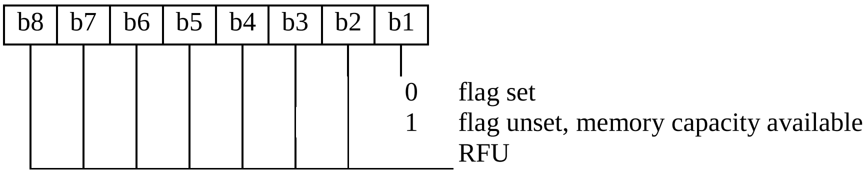
As defined in ETS 300 392-2[2].

SDS4 “Memory capacity exceeded” notification flag

Contents:

This flag is required to allow a process of flow control, so that as memory capacity becomes available, the network service centre can be informed.

Coding:



SDS123 “memory capacity exceeded" notification flag

same as SDS4 “memory capacity exceeded"

SDS4 last used message index:

Contents:

The value of the last message index used for the SDS4 message.

Coding: two bytes

SDS123 last used message index:

Contents:

The value of the last message index used for the SDS123 message.

Coding: two bytes

10.3.55 EF$_{\text{WELCOME}}$ (Welcome Message)

This EF contains an alpha-numeric message displayed during the ME boot sequence.

<table>
<tr><td colspan="2">Identifier: 6F37'</td><td colspan="2">Structure: transparent</td><td>Optional</td></tr>
<tr><td colspan="3">File size: 32 bytes</td><td colspan="2">Update activity: low</td></tr>
<tr><td colspan="5">Access Conditions:
READ CHV1
UPDATE ADM
INVALIDATE ADM
REHABILITATE ADM</td></tr>
<tr><td>Bytes</td><td colspan="2">Description</td><td>M/O</td><td>Length</td></tr>
<tr><td>1 - 32</td><td colspan="2">Message string</td><td>M</td><td>32 bytes</td></tr>
</table>

Message string

Contents:

A string defined by the network operator

Coding:

According to the default 8-bit alphabet ISO 8859-1 [24]. Unused bytes shall be set as 'FF'.

10.3.56 EF$_{\text{SDSR}}$ (SDS delivery report)

This EF contains information in accordance with EN 300 392-2 [2] comprising delivery report messages which have been received by the MS from the network.

Each record is used to store the delivery report of a short data service message. The first byte of each record is the link between the delivery report and the corresponding SDS in EF$_{\text{SDS4}}$.

Identifier: '6F38'	Structure: linear fixed	Optional
Record length: 2 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	SDS record identifier	M	1
2	SDS delivery status	M	1

SDS record identifier

Contents:

This data idem identifies the corresponding SDS record in EF$_{\text{SDS4,}}$ e.g. if this byte is coded '05' then this delivery report corresponds to the SDS record #5 of EF$_{\text{SDS4.}}$

Coding:

'00' empty record

'01' – 'FF' record number of the corresponding SDS in EF$_{\text{SDS4.}}$

SDS delivery status:

This data item contains the delivery status as defined in EN 300 392-2 [2].

10.3.57 EF$_{\text{SDSP}}$ (SDS parameters)

This EF contains values for short data service header parameters, which can be used by the ME for user assistance in preparation of mobile originated SDS.

The EF consists of one or more records, with each record able to hold a set of SDS parameters. The first record in the EF shall be used as a default set of parameters, if no other record is selected.

To distinguish between records, an alpha identifier is included within each record, coded on X bytes.

Identifier: '6F39 '	Structure: linear fixed	Optional
Record size: X+19 bytes	Update activity: low	

Access Conditions:

READ	CHV1
UPDATE	CHV1
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to X	Alpha identifier	M	X bytes
X+1	Parameter indicators	M	1 byte
X+2 to X+16	Service centre address	M	15 bytes
X+17	Protocol identifier	M	1 byte
X+18	Data coding scheme	M	1 byte
X+19	Validity period	M	1 byte

Storage is allocated for all the possible SDS parameters, regardless of whether they are present or absent. Any bytes unused, due to parameters not requiring all of the bytes, or due to absent parameters, shall be set to 'FF'.

Alpha identifier

Contents:

Alpha tag of the associated SDS – parameter

Coding:

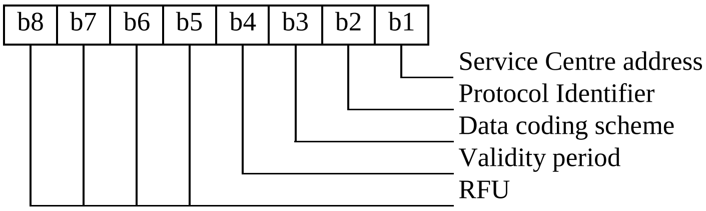
As defined in 10.4.1.

Parameter Indicators

Contents:

Each of the default SDS parameters which can be stored in the remainder of the record are marked absent or present by individual bits within this byte.

Coding:



Bit value: 0 – parameter present

1 – parameter absent

Service centre address:

Contents:

Service centre address

Coding:

As defined for the message destination/source identifier in clause 10.3.42

Protocol Identifier

As defined for the protocol identifier in clause 10.3.42

Data coding scheme

As defined in EN 300 392-2 [2]

Validity period

As defined in EN 300 392-2 [2]

10.3.58 EF$_{DIALSC}$ (Dialling schemes for TETRA network)

This EF contains the information indicating the dialling scheme.

Identifier: '6F46'	Structure: transparent	Mandatory
File length: 5 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Type of dialling	M	1
2	Number of digits	M	1
3 to 5	Base address	M	3

Type of dialling:

Contents: the type of dialling scheme to be selected.

Coding:

b8	b7	b6	b5	b4	b3	b2	b1	
						0	0	ISSI or ITSI dialling
						0	1	FSSN Dialling
						1	0	RFU
						1	1	RFU
								RFU

Number of digits

Contents:

In case of FSSN dialling, up to this number of digits, the number dialled has to be added to the base address. Else the dialling is as ISSI/ITSI dialling.

Coding: 1 byte

“FF” in case of ISSI/ITSI dialling, else number of digits.

Base Address

Contents:

It contains the base address to which the dialled number has to be added.

Coding:

3 bytes – used in case of FSSN dialling else set to “FF FF FF”

10.3.59 EF_{APN} (APN table)

This EF contains a list of APNs (IP access point names) which the ME can use to match the access point name string to the corresponding index which is used in the air interface (EN 300 392-2 [2]).

Identifier: '6F3E '	Structure: linear fixed		Optional
Record size: 65 bytes		Update activity: high	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			

Bytes	Description	M/O	Length
1-2	Access point name index	M	2
3-65	Access point name	M	63

Access point name index:

Contents:

The Access point name index is used over the air interface.

Coding:

The message value is coded with two bytes as defined in EN 300 392-2 [2]

Access point name:

Contents:

The alphanumeric name the user the has assigned for corresponding access point name index.

Coding:

According to the default 8-bit alphabet ISO 8859-1 [24].

NOTE: The access point name stored in this EF does not have to be the same as the access point name sent by TETRA SwMI towards the IP gateway. This is because only the access point name index is sent over the air interface. The SwMI maps the index to the real APN Network Identifier that is sent to the GGSN network element (GSM 03.03 [13]).

10.3.60 EF_{PNI} (Private Number Information)

Each record of this EF contains a number structure definition and stores the user's own private number. The number structure definition allows the MS to understand the structure of different Private Number Plans that may be in use. This enables the MS to display the user's own private number correctly.

The first record contains the default private number information, the other records are in descending order of priority.

The selection of which type of Private Number Plan to use is outside the scope of the present document.

Identifier: '6F C0'	Structure: linear fixed		Optional
Record length: 14 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1/CHV2 INVALIDATE ADM REHABILITATE ADM			

Bytes	Description	M/O	Length
1 to 2	Tier Details	M	2
3 to 14	Private Number	M	12

Tier Details

Contents:

This field of each record defines the hierarchical structure of the private number, allowing up to four variable length tiers in descending order of significance.

Coding:

The tier lengths are binary encoded nibbles.

The number of tiers in the hierarchy is N, where N may take the value 1 to 4.

There is no absolute hierarchy, the structure is relative. For example if there are two tiers in the hierarchy the first two tier fields (N and N-1) are set to the length of digits in each, the remaining two tiers (N-2 and N-3) will be set to '0'.

'00 00' - No Private Number Stored

'01 mn' signifies that what follows is concatenation of m digit leading number + n digit second number + [remainder] with unused digits padded with 'F'.

e.g. the full coding for an FSSN number "ab cdef" with 2+4 structure might be:

'01 02 ab cd ef FF FF FF FF FF FF FF FF'

or the full coding for a private number "ab cdefg hijk" with 2+5+4 structure might be:

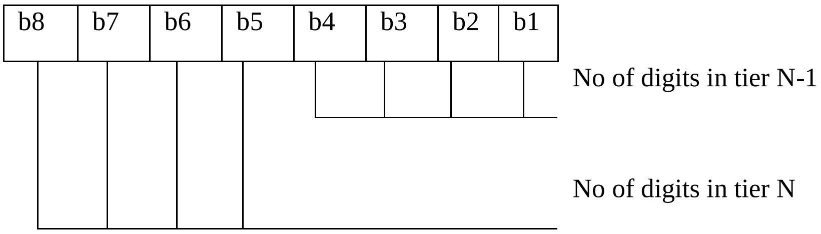
'01 25 ab cd ef hijk FF FF FF FF FF FF FF'

'01 FF' – 1-24 digit private number with no tier structure defined

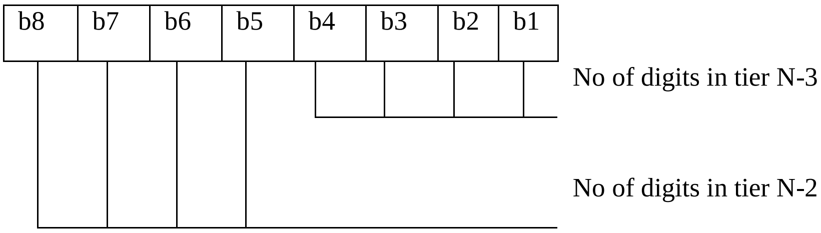
'XX XX' – 1 to 4 tier Private number stored (where X takes the range '1' to 'F' hex and the sum of digits does not exceed 24

'FF FF' – No valid number follows.

Byte 1



Byte 2



Private Number

Contents:

This field of each record allows storage of a private number

Coding:

A contiguous string of left-justified BCD encoded digits, starting with the most significant digit. Where the number is shorter than 24 digits the remaining digits shall be padded with 'F'.

10.4 Contents of the EFs at the Telecom level

10.4.1 EF$_{\text{ADN}}$ (Abbreviated dialling numbers)

This EF contains Abbreviated Dialling Numbers (ADN) and/or Supplementary Service Control strings (SSC). In addition it contains identifiers of associated network/bearer capabilities and identifiers of extension records. It may also contain an associated alpha-tagging.

Identifier: '6F3A'	Structure: linear fixed		Optional
Record length: X+14 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE CHV2 REHABILITATE CHV2			
Bytes	Description	M/O	Length
1 to X	Alpha Identifier	O	X bytes
X+1	Length of BCD number/SSC contents	M	1 byte
X+2	TON and NPI	M	1 byte
X+3 to X+12	Dialling Number/SSC String	M	10 bytes
X+13	Capability/Configuration Identifier	M	1 byte
X+14	Extension1 Record Identifier	M	1 byte

Alpha Identifier

Contents:

Alpha-tagging of the associated dialling number.

Coding:

this alpha-tagging shall use either

- the SMS default 7-bit coded alphabet as defined in GSM 03.38 [10] with bit 8 set to 0. The alpha identifier shall be left justified. Unused bytes shall be set to 'FF'.

or

- one of the UCS2 coded options as defined in GSM 11.11 [9].

NOTE 1: The value of X may be from zero to 241. Using the command GET RESPONSE the ME can determine the value of X.

Length of BCD number/SSC contents

Contents:

this byte gives the number of bytes of the following two data items containing actual BCD number/SSC information. This means that the maximum value is 11, even when the actual ADN/SSC information length is greater than 11. When an ADN/SSC has extension, it is indicated by the extension1 identifier being unequal to 'FF'. The remainder is stored in the EF_{EXT1} with the remaining length of the additional data being coded in the appropriate additional record itself (see clause 10.4.6).

Coding:

according to GSM 04.08 [14].

TON and NPI

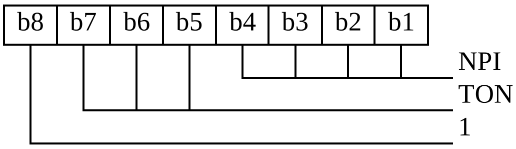
Contents:

Type of number (TON) and numbering plan identification (NPI).

Coding:

according to GSM 04.08 [14]. If the Dialling Number/SSC String does not contain a dialling number, e.g. a control string deactivating a service, the TON/NPI byte shall be set to 'FF' by the ME (see note 2).

NOTE 2: If a dialling number is absent, no TON/NPI byte is transmitted over the radio interface (see GSM 04.08 [14]). Accordingly, the ME should not interpret the value 'FF' and not send it over the radio interface.



Dialling Number/SSC String

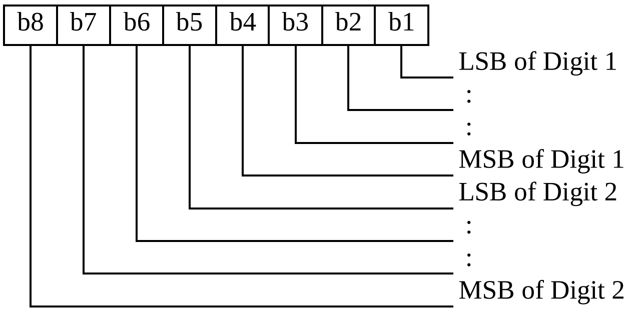
Contents:

up to 20 digits of the telephone number and/or SSC information.

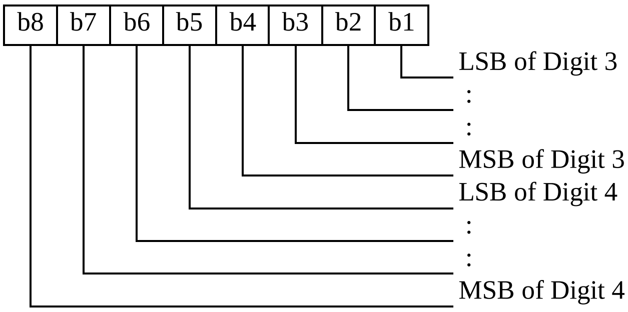
Coding:

according to GSM 04.08 [14], GSM 02.30 [12] and the extended BCD-coding (see table 12). If the telephone number or SSC is longer than 20 digits, the first 20 digits are stored in this data item and the remainder is stored in an associated record in the EF_{EXT1}. The record is identified by the Extension1 Record Identifier. If ADN/SSC require less than 20 digits, excess nibbles at the end of the data item shall be set to 'F'. Where individual dialled numbers, in one or more records, of less than 20 digits share a common appended digit string the first digits are stored in this data item and the common digits stored in an associated record in the EF_{EXT1}. The record is identified by the Extension 1 Record Identifier. Excess nibbles at the end of the data item shall be set to 'F'.

Byte X+3



Byte X+4:



etc.

Capability/Configuration Identifier

Contents:

capability/configuration identification byte. This byte identifies the number of a record in the EF_{CCP} containing associated capability/configuration parameters required for the call. The use of this byte is optional. If it is not used it shall be set to 'FF'.

Coding:

binary.

Extension1 Record Identifier

Contents:

extension1 record identification byte. This byte identifies the number of a record in the EF_{EXT1} containing an associated called party subaddress or additional data. The use of this byte is optional. If it is not used it shall be set to 'FF'.

If the ADN/SSC requires both additional data and called party subaddress, this byte identifies the additional record. A chaining mechanism inside EF_{EXT1} identifies the record of the appropriate called party subaddress (see clause 10.4.6).

Coding: binary.

NOTE 3: As EF_{ADN} is part of the $DF_{TELECOM}$ it may be used by TETRA and also other applications in a multi-application card. If the non-GSM application does not recognize the use of Type of Number (TON) and Number Plan Identification (NPI), then the information relating to the national dialling plan must be held within the data item dialling number/SSC and the TON and NPI fields set to UNKNOWN. This format would be acceptable for GSM operation and also for the non-GSM application where the TON and NPI fields shall be ignored.

EXAMPLE: SIM storage of an International Number using ITU-T-E.164 [28] numbering plan

	TON	NPI	Digit field
GSM application	001	0001	abc...
Other application compatible with GSM	000	0000	xxx...abc...

where "abc..." denotes the subscriber number digits (including its country code), and "xxx..." denotes escape digits or a national prefix replacing TON and NPI.

NOTE 4: When the ME acts upon the EF_{ADN} with a SEEK command in order to identify a character string in the alpha-identifier, it is the responsibility of the ME to ensure that the number of characters used as SEEK parameters are less than or equal to the value of X if the MMI allows the user to offer a greater number.

Table 7: Extended BCD coding

BCD Value	Character/Meaning
'0'	"0"
...	...
'9'	"9"
'A'	"*"
'B'	"#"
'C'	DTMF Control digit separator (GSM 02.07 [11])
'D'	"Wild" value This will cause the MMI to prompt the user for a single digit (see GSM 02.07 [11]).
'E'	Expansion digit ("Shift Key"). It has the effect of adding '10' to the following digit. The following BCD digit will hence be interpreted in the range of '10'-'1E'. The purpose of digits in this range is for further study.
'F'	Endmark e.g. in case of an odd number of digits

BCD values 'C', 'D' and 'E' are never sent across the radio interface.

NOTE 5: The interpretation of values 'D', 'E' and 'F' as DTMF digits is for further study.

NOTE 6: A second or subsequent 'C' BCD value will be interpreted as a 3 second PAUSE (see GSM 02.07 [11]).

10.4.2 EF$_{FDN}$ (Fixed dialling numbers)

This EF contains Fixed Dialling Numbers (FDN) and/or Supplementary Service Control strings (SSC). In addition it contains identifiers of associated network/bearer capabilities and identifiers of extension records. It may also contain an associated alpha-tagging.

Identifier: '6F3B'	Structure: linear fixed		Optional
Record length: X+14 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV2 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1 to X	Alpha Identifier	O	X bytes
X+1	Length of BCD number/SSC contents	M	1 byte
X+2	TON and NPI	M	1 byte
X+3 to X+12	Dialling Number/SSC String	M	10 bytes
X+13	Capability/Configuration Identifier	M	1 byte
X+14	Extension2 Record Identifier	M	1 byte

For contents and coding of all data items see the respective data items of the EF$_{ADN}$ (clause 10.4.1), with the exception that extension records are stored in the EF$_{EXT2}$.

NOTE: The value of X (the number of bytes in the alpha-identifier) may be different to the length denoted X in EF$_{ADN}$.

10.4.3 EF_{MSISDN} (MSISDN)

This EF contains MSISDN(s) related to the subscriber. In addition it contains identifiers of associated network/bearer capabilities and identifiers of extension records. It may also contain an associated alpha-tagging.

Identifier: '6F40'	Structure: linear fixed		Optional
Record length: X+14 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1 to X	Alpha Identifier	O	X bytes
X+1	Length of BCD number/SSC contents	M	1 byte
X+2	TON and NPI	M	1 byte
X+3 to X+12	Dialling Number/SSC String	M	10 bytes
X+13	Capability/Configuration Identifier	M	1 byte
X+14	Extension1 Record Identifier	M	1 byte

For contents and coding of all data items see the respective data items of EF_{ADN}.

NOTE 1: If the SIM stores more than one MSISDN number and the ME displays the MSISDN number(s) within the initialization procedure then the one stored in the first record shall be displayed with priority.

NOTE 2: The value of X (the number of bytes in the alpha-identifier) may be different to the length denoted X in EF_{ADN}.

10.4.4 EF_{LND} (Last number dialled)

This EF contains the last numbers dialled (LND) and/or the respective supplementary service control strings (SSC). In addition it contains identifiers of associated network/bearer capabilities and identifiers of extension records. It may also contain associated alpha-tagging.

Identifier: '6F44'	Structure: cyclic		Optional
Record length: X+14 bytes		Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INCREASE NEVER INVALIDATE ADM REHABILITATE ADM			
Bytes	Description	M/O	Length
1 to X	Alpha Identifier	O	X bytes
X+1	Length of BCD number/SSC contents	M	1 byte
X+2	TON and NPI	M	1 byte
X+3 to X+12	Dialling Number/SSC String	M	10 bytes
X+13	Capability/Configuration Identifier	M	1 byte
X+14	Extension1 Record Identifier	M	1 byte

For contents and coding, see clause 10.4.1 EF_{ADN}.

The value of X in EF_{LND} may be different to both the value of X in EF_{ADN} and of X in EF_{FDN}.

If the value of X in $EF_{LND}$is longer than the length of the α-tag of the number to be stored, then the ME shall pad the α-tag with 'FF'. If the value of X in EF_{LND} is shorter than the length of the α-tag of the number to be stored, then the ME shall cut off excessive bytes.

10.4.5 EF$_{\text{SDN}}$ (Service Dialling Numbers)

This EF contains special service numbers (SDN) and/or the respective supplementary service control strings (SSC). In addition it contains identifiers of associated network/bearer capabilities and identifiers of extension records. It may also contain associated alpha-tagging.

Identifier: '6F49'	Structure: linear fixed	Optional
Record length: X+14 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-X	Alpha identifier	O	X bytes
X+1	Length of BCD number/SSC contents	M	1 bytes
X+2	TON and NPI	M	1 byte
X+3-X+12	Dialling Number/SSC String	M	10 bytes
X+13	Capability/Configuration Identifier	M	1 byte
X+14	Extension3 Record Identifier	M	1 byte

For contents and coding of all data items see the respective data items of the EF$_{\text{ADN}}$ (clause 10.4.1), with the exception that extension records are stored in the EF$_{\text{EXT3}}$.

NOTE: The value of X (the number of bytes in the alpha-identifier) may be different to the length denoted X in EF$_{\text{ADN}}$.

10.4.6 EF$_{\text{EXT1}}$ (Extension1)

This EF contains extension data of an ADN/SSC, an MSISDN, or an LND. Extension data is caused by:

- an ADN/SSC (MSISDN, LND) which is greater than the 20 digit capacity of the ADN/SSC (MSISDN, LND) Elementary File or where common digits are required to follow an ADN/SSC string of less than 20 digits. The remainder is stored in this EF as a record, which is identified by a specified identification byte inside the ADN/SSC (MSISDN, LND) Elementary File. The EXT1 record in this case is specified as additional data;
- an associated called party subaddress. The EXT1 record in this case is specified as subaddress data.

Identifier: '6F4A'	Structure: linear fixed	Optional
Record length: 13 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV1 INVALIDATE ADM REHABILITATE ADM		

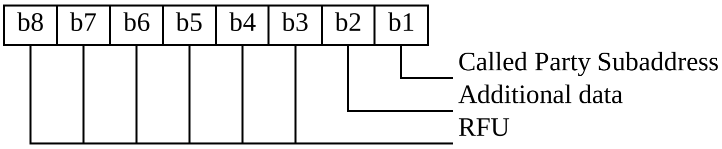
Bytes	Description	M/O	Length
1	Record type	M	1 byte
2 to 12	Extension data	M	11 bytes
13	Identifier	M	1 byte

Record type

Contents:

Type of the record

Coding:



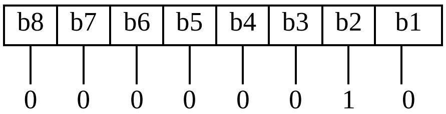
b3-b8 are reserved and set to 0;

a bit set to 1 identifies the type of record;

only one type can be set;

'00' indicates the type "unknown".

The following example of coding means that the type of extension data is "additional data":



Extension data

Contents:

Additional data or Called Party Subaddress depending on record type.

Coding:

Case 1, Extension1 record is additional data:

The first byte of the extension data gives the number of bytes of the remainder of ADN/SSC (respectively MSISDN, LND). The coding of remaining bytes is BCD, according to the coding of ADN/SSC (MSISDN, LND). Unused nibbles at the end have to be set to 'F'. It is possible if the number of additional digits exceeds the capacity of the additional record to chain another record inside the EXT1 Elementary File by the identifier in byte 13.

Case 2, Extension1 record is Called Party Subaddress:

The subaddress data contains information as defined for this purpose in GSM 04.08 [14]. All information defined in GSM 04.08 [14], except the information element identifier, shall be stored in the SIM. The length of this subaddress data can be up to 22 bytes. In those cases where two extension records are needed, these records are chained by the identifier field. The extension record containing the first part of the called party subaddress points to the record which contains the second part of the subaddress.

Identifier

Contents:

identifier of the next extension record to enable storage of information longer than 11 bytes.

Coding:

record number of next record. 'FF' identifies the end of the chain.

EXAMPLE: A chain of extension records being associated to an ADN/SSC. The extension1 record identifier (Byte 14+X) of ADN/SSC is set to 3.

No of Record	Type	Extension Data	Next	Record
:	:	:	:	
:	:	:	:	
Record 3	'02'	xx xx	'06'	
Record 4	'xx'	xx xx	'xx'	
Record 5	'01'	xx xx	'FF'	
Record 6	'01'	xx xx	'05'	
:	:	:	:	
:	:	:	:	

In this example ADN/SSC is associated to additional data (record 3) and a called party subaddress whose length is more than 11 bytes (records 6 and 5).

10.4.7 EF_{EXT2} (Extension2)

This EF contains extension data of an FDN/SSC (see EXT2 in 10.4.2).

Identifier: '6F4B'	Structure: linear fixed	Optional
Record length: 13 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE CHV2 INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Record type	M	1 byte
2 to 12	Extension data	M	11 bytes
13	Identifier	M	1 byte

For contents and coding see clause 10.4.6 EF_{EXT1}.

10.4.8 EF_{EXT3} (Extension3)

This EF contains extension data of an SDN (see EXT3 in 10.4.5).

Identifier: '6F4C'	Structure: linear fixed	Optional
Record length: 13 bytes	Update activity: low	
Access Conditions: READ CHV1 UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Record type	M	1 byte
2 to 12	Extension data	M	11 bytes
13	Identifier	M	1 byte

For contents and coding see clause 10.4.6 EF_{EXT1}.

10.5 Files of TETRA

This clause contains a figure depicting the file structure of the SIM. DF$_{TETRA}$ shall be selected by using the identifier '7F90'.

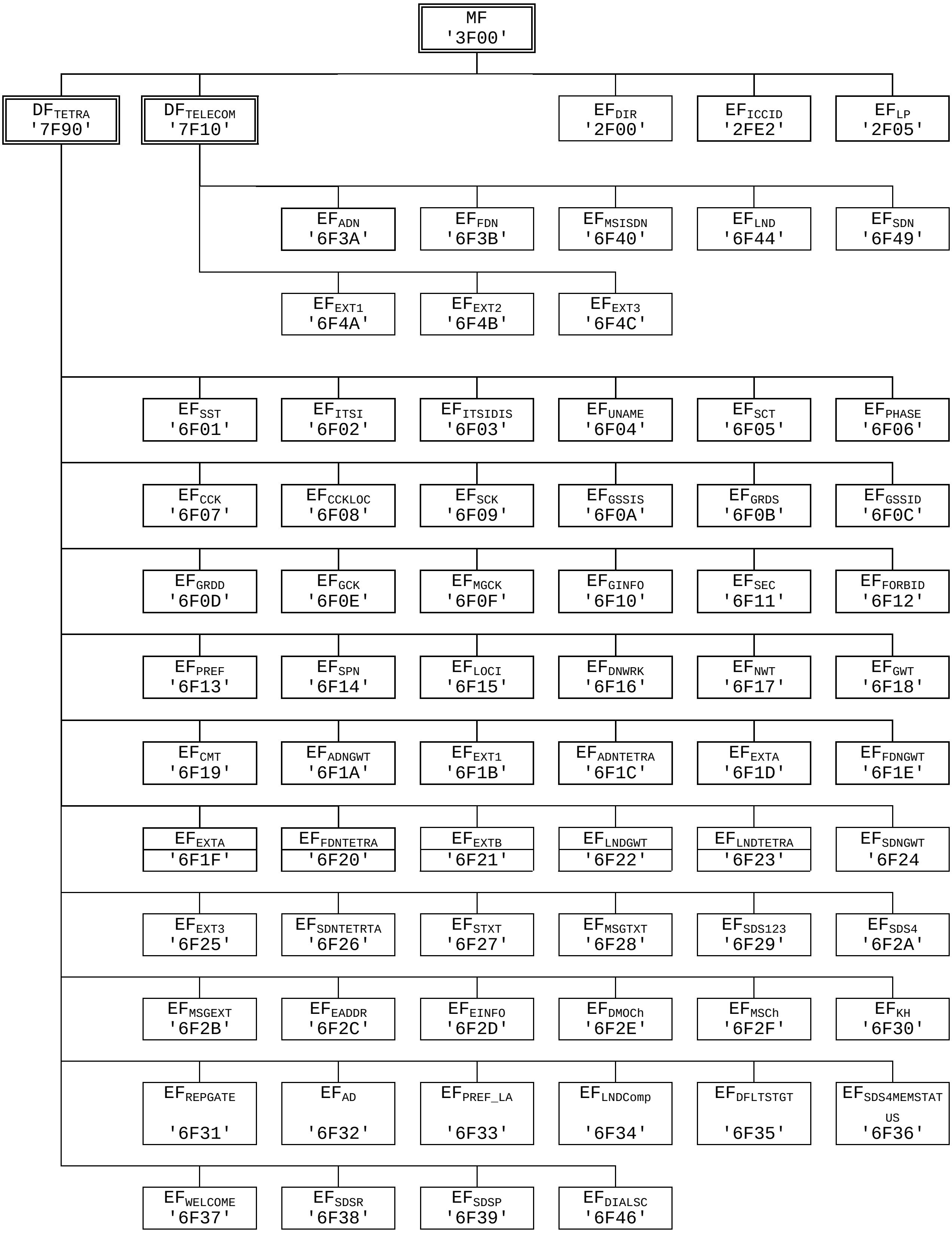


Figure 7: File identifiers and directory structures of TETRA

11 Application protocol

The SIM interfaces with appropriate terminal equipment(ME) when in TETRA administrative mode. These operations are outside the scope of the present document.

During TETRA network operations, the SIM exchanges messages with the ME via the SIM/ME interface.

A message can be a command or a response as follows:

- a TETRA command/response pair is a sequence consisting of a command and the associated response;
- a TETRA procedure consists of one or more TETRA command/response pairs which are used to perform all or part of an application-oriented task. A procedure shall be considered as a whole, that is to say that the corresponding task is achieved if and only if the procedure is completed. The ME shall ensure that, when operated according to the manufacturer's manual, any unspecified interruption of the sequence of command/response pairs which realize the procedure, leads to the abortion of the procedure itself;
- a TETRA session of the SIM in the TETRA application is the interval of time starting at the completion of the SIM initialization procedure and ending either with the start of the TETRA session termination procedure, or at the first instant the link between the SIM and the ME is interrupted.

During the TETRA network operation phase, the ME plays the role of the master and the SIM plays the role of the slave.

The list of procedures at the SIM/ME interface in TETRA network operation are listed in the following table:

The ME automatically initiates some procedures. They are marked “ME”.

NOTE1: Some procedures at the SIM/ME interface require MMI interactions. The following descriptions do not intend to infer any specific implementation of the corresponding MMI. When MMI interaction is required, it is marked "MMI".

NOTE 2: Some procedures are not clearly user dependent. They are directly caused by the interaction of the MS and the network. Such procedures are marked NETwork "(NET)".

General Procedures:

Reading an EF	ME;
Updating an EF	ME.

SIM management procedures:

SIM initialization	ME;
TETRA session initialization	ME;
TETRA session termination	ME;
Language preference request	ME;
Administrative information request	ME;
SIM service table request	ME;
SIM phase request	ME;
SIM presence detection	ME.

CHV related procedures:

CHV verification	MMI;
CHV value substitution	MMI;
CHV disabling	MMI;
CHV enabling	MMI;
CHV unblocking	MMI.

TETRA security related procedures:

TETRA algorithms computation	NET;
TETRA key computation (SCK, DCK, MGCK, GCK)	NET;
ITSI request	NET;
ITSI disabling	NET;
Location Information	NET;
Broadcast network information	NET;
Forbidden networks information	NET.

Subscription related procedures:

Username	MMI;
Subscriber class request	ME;
Group information	MMI/NET;
User's group information	ME/NET;
Call modifiers	NET/ME;
Network information	ME;
Dialling Numbers (ADN, ADNTETRA, ADNGWT, FDN, FDNTETRA, FDNGWT, LND, LNDTETRA, LNDGWT, SDN, SDNTETRA, SDNGWT LNDComp)	MMI/ME;
SDS messages (Message texts, SDS123 and SDS4)	MMI;
Preferred networks	MMI;
Service Provider Name (SPN)	ME;
ICCID	ME;
Emergency addresses	ME/MMI;

11.1 General procedures

11.1.1 Reading an EF

The ME selects the EF and sends a READ command. This contains the location of the data to be read. If the access condition for READ is fulfilled, the SIM sends the requested data contained in the EF to the ME. If the access condition is not fulfilled, no data will be sent and an error code will be returned.

11.1.2 Updating an EF

The ME selects the EF and sends an UPDATE command. This contains the location of the data to be updated and the new data to be stored. If the access condition for UPDATE is fulfilled, the SIM updates the selected EF by replacing the existing data in the EF with that contained in the command. If the access condition is not fulfilled, the data existing in the EF will be unchanged, the new data will not be stored, and an error code will be returned.

In some cases, files are updated by running an algorithm resident on the SIM.

11.1.3 Invalidating an EF

The ME selects the EF and sends an INVALIDATE command. If the access conditions of INVALIDATE are fulfilled the EF is invalidated.

11.2 SIM management procedures

The procedures listed in this clause are required for execution of the procedures in clause 113,3, 11,4 and 11.5

11.2.1 SIM initialization

The ME runs the language request procedure. If none of the indicated languages are available, the ME selects a default language (e.g. English).

The ME checks the presence of a EF_{CHV1} at master file level. If the read access condition is CHV, the ME runs the CHV verification procedure for CHV1 as defined in clause 11.3.1.

11.2.2 TETRA session initialization

Following the SIM initialization, the ME selects DF_{TETRA} by using the identifier or by the path given in $EF_{DIR.}$ The ME then selects EF_{ITSI} to obtain its INVALIDATION status. If the ITSI is invalidated the ME informs the user and the TETRA session initialization fails.

The ME runs the CHV verification procedure for CHV1 as defined in clause 11.3.1.. If the CHV verification is unsuccessful, the TETRA session initialization fails.

NOTE: If there is no EF_{CHV1} present at the application level, there has to be one at the master file level. For convenience of the user, implementations having both an EF_{CHV} at application and at master file level should be avoided.

If the CHV verification procedure is performed successfully, the ME then runs the following procedures:

- Administrative information request;
- SIM Phase request;
- SIM Service Table request;
- ITSI request;
- ITSI temporarily disabled enquiry;
- Subscriber class request;
- Preferred networks request;
- Location Information request;
- Mutual authentication requirement request;
- Forbidden networks request;
- Interrupted emergency call request.

After the SIM initialization has been completed successfully, the MS is ready for a TETRA session.

NOTE: If the ITSI is "Temporary disabled by SwMI", the ME enters a TETRA session with a restricted mode of operation. The restricted TETRA session usually consists of the MS simply listening to the SwMI to eventually detect a re-enabling of the ITSI by the network (see EN 300 392-7 [3]).

11.2.3 TETRA session termination

NOTE 1: This procedure is not to be confused with the deactivation procedure in clause 4.3.2.

The ME terminates the TETRA session as follows:

The ME runs all the procedures that are necessary to transfer the following subscriber related information to the SIM:

As soon as the SIM indicates that these procedures are completed, the ME/SIM link may be deactivated.

Finally, the ME deletes all these subscriber related information elements from its memory.

NOTE 2: If the ME has already updated any of the subscriber related information during the TETRA Session, and the value has not changed until TETRA session termination, the ME may omit the respective update procedure.

11.2.4 Language preference request

Request: The ME performs the reading procedure with EF_{LP}.

Update: The ME performs the updating procedure with EF_{LP}.

11.2.5 Administrative information request

Request: The ME performs the reading procedure with EF_{AD}.

Update: The ME performs the updating procedure with $EF_{AD.}$

11.2.6 SIM service table request

The ME performs the reading procedure with EF_{SST}.

11.2.7 SIM phase request

The ME performs the reading procedure with $EF_{PHASE.}$

11.2.8 SIM presence detection

As an additional mechanism, to ensure that the SIM has not been removed during a card session, the ME sends, at frequent intervals, a STATUS command during each call. A STATUS command shall be issued within all 30 second periods of inactivity on the SIM-ME interface during a call. Inactivity in this case is defined as starting at the end of the last communication or the last issued STATUS command. If no response data is received to this STATUS command, then the call shall be terminated as soon as possible but at least within 5 seconds after the STATUS command has been sent. If the DF indicated in response to a STATUS command is not the same as that which was indicated in the previous response, or accessed by the previous command, then the call shall be terminated as soon as possible but at least within 5 seconds after the response data has been received. This procedure shall be used in addition to a mechanical or other device used to detect the removal of a SIM.

11.2.9 SIM card number request

The ME performs the reading procedure with $EF_{ICCID.}$

11.2.10 Common Cipher Key request

The ME performs the read procedure with EF_{CCK} to obtain the current record in this EF.

11.3 CHV related procedures

The procedures listed in this clause are mandatory

A successful completion of one of the following procedures grants the access right of the corresponding CHV for the TETRA session. This right is valid for all files within the application(s) protected by this CHV.

After a third consecutive presentation of a wrong CHV to the SIM, not necessarily in the same TETRA session, the CHV status becomes "blocked" and the access right previously granted by this CHV is lost immediately.

An access right is not granted if any of the following procedures are unsuccessfully completed or aborted.

11.3.1 CHV verification

The ME checks the CHV status.

In the case of CHV1 the following procedures applies:

If the CHV1 status is "blocked", and CHV1 is “enabled” the procedure ends and is finished unsuccessfully.

if the CHV1 status is "blocked" but CHV1 is "disabled", the procedure ends and is finished successfully. The ME shall, however, accept SIMs which do not grant access rights when CHV1 is "blocked" and "disabled". In that case ME shall consider those SIMs as "blocked";

If the CHV1 status is not "blocked", but CHV1 is "disabled", the procedure is finished successfully.

If the CHV1 status is not "blocked" and CHV1 is "enabled", the ME uses the VERIFY CHV1 function. If the CHV1 presented by the ME is equal to the corresponding CHV1 stored in the SIM, the procedure is finished successfully. If the CHV1 presented by the ME is not equal to the corresponding CHV 1stored in the SIM, the procedure ends and is finished unsuccessfully.

In the case of CHV2 the following procedure applies:

if the CHV2 status is "blocked", the procedure ends and is finished unsuccessfully;

if the CHV2 status is not "blocked", the ME uses the VERIFY CHV function. If the CHV2 presented by the ME is equal to the corresponding CHV2 stored in the SIM, the procedure is finished successfully. If the CHV2 presented by the ME is not equal to the corresponding CHV2 stored in the SIM, the procedure ends and is finished unsuccessfully.

11.3.2 CHV value substitution

The ME checks the CHV status. If the CHV status is "blocked" or "disabled", the procedure ends and is finished unsuccessfully.

If the CHV status is not "blocked" and the enabled/disabled indicator is set "enabled", the ME uses the CHANGE CHV function. If the old CHV presented by the ME is equal to the corresponding CHV stored in the SIM, the new CHV presented by the ME is stored in the SIM and the procedure is finished successfully.

If the old CHV and the CHV in memory are not identical, the procedure ends and is finished unsuccessfully.

11.3.3 CHV disabling

Requirement: Service no.1 "available".

The ME checks the CHV1 status. If the CHV1 status is "blocked", the procedure ends and is finished unsuccessfully.

If the CHV1 status is not "blocked", the ME reads the CHV1 enabled/disabled indicator. If this is set "disabled", the procedure ends and is finished unsuccessfully.

If the CHV1 status is not "blocked" and the enabled/disabled indicator is set "enabled", the ME uses the DISABLE CHV function. If the CHV1 presented by the ME is equal to the CHV1 stored in the SIM, the status of CHV1 is set "disabled" and the procedure is finished successfully. If the CHV1 presented by the ME is not equal to the CHV1 stored in the SIM, the procedure ends and is finished unsuccessfully.

This requirement applies to the CHV1 at the TETRA application level. For the CHV1 at the master file level, it only applies in the case of a TETRA only card.

11.3.4 CHV enabling

The ME checks the CHV1 status. If the CHV1 status is "blocked", the procedure ends and is finished unsuccessfully.

If the CHV1 status is not "blocked", the ME reads the CHV1 enabled/disabled indicator. If this is set "enabled", the procedure ends and is finished unsuccessfully.

If the CHV1 status is not "blocked" and the enabled/disabled indicator is set "disabled", the ME uses the ENABLE CHV function. If the CHV1 presented by the ME is equal to the CHV1 stored in the SIM, the status of CHV1 is set "enabled" and the procedure is finished successfully. If the CHV presented by the ME is not equal to the CHV1 stored in the SIM, the procedure ends and is finished unsuccessfully.

11.3.5 CHV unblocking

The execution of the CHV unblocking procedure is independent of the corresponding CHV status, i.e. being blocked or not.

The ME checks the UNBLOCK CHV status. If the UNBLOCK CHV status is "blocked", the procedure ends and is finished unsuccessfully.

If the UNBLOCK CHV status is not "blocked", the ME uses the UNBLOCK CHV function. If the UNBLOCK CHV presented by the ME is equal to the corresponding UNBLOCK CHV stored in the SIM, the relevant CHV status becomes "unblocked" and the procedure is finished successfully. If the UNBLOCK CHV presented by the ME is not equal to the corresponding UNBLOCK CHV stored in the SIM, the procedure ends and is finished unsuccessfully.

11.4 TETRA security related procedures

The procedures listed in this clause are only executable if the associated services, which are optional, are provided in the SIM. However, if the procedures are implemented, they shall be in accordance with the requirement stated in this clause. If a procedure is related to a specific service indicated in the SIM service table, it shall only be executed if the corresponding bit denoting this service as "available" (see EF_{SST}). In all other cases this procedure shall not start.

The SIM security procedures are associated with the air interface message exchange protocol procedures for authenticating the SIM to a TETRA network and the TETRA network to the SIM. During these SIM security procedures the card runs the specified algorithms TA11/12 and TA21/22 to calculate respectively the expected response from the SIM, (X)RES1 with its associated derived cipher key DCK1 and the expected response from the SwMI, (X)RES2 with its associated derived cipher key DCK2.

On successful authentication the derived cipher key DCK, used for encrypting air interface signalling and traffic channels, shall be derived from its two parts DCK1 and DCK2 by running the TB4 algorithm.

All the algorithms shall not be executable unless DF_{TETRA} has been selected as the Current Directory and a successful CHV verification procedure has been performed (see clause 11.3.1).

The procedures are either initiated by the ME (internal applications or MMI) or interfaced from the SwMI via the ME. In the latter case the ME provides only a delivery service with no other functionality than to interpret the PDUs if necessary.

11.4.1 Authentication procedures and generation of DCK

11.4.1.1 Mutual authentication requirement request

The SIM performs the read procedure with EF_{SEC} to determine whether a mutual authentication is requested by the SIM in case of a SIM authentication request from the SwMI.

11.4.1.2 SIM authentication

The ME runs the TA11/12 ALGORITHM, followed by a GET RESPONSE to obtain the RES1. If and only if the SIM requests a mutual authentication (see clause 11.4.1.1), the ME runs then the GET RANDOM, followed by the TA21/22 ALGORITHM. If the authentication was successful, it finally runs the TB4 ALGORITHM to obtain DCK.

11.4.1.3 SwMI authentication

The ME runs the GET RANDOM function, followed by the TA21/22 ALGORITHM. If and only if the SwMI requests a mutual authentication, the ME runs the TA11/12 ALGORITHM, followed by a GET RESPONSE to obtain the RES1. If the authentication was successful, it finally runs the TB4 ALGORITHM to obtain DCK.

11.4.2 TETRA OTAR key computation (CCK, GCK, SCK)

The CCK, GCK and SCK cipher keys can be updated by OTAR. They are sent over the air interface in sealed format and need to be unsealed on receipt by algorithms on the SIM.

SCK and CCK are accessible from the SIM-ME interface but GCK is accessible only in modified format (MGCK).

11.4.2.1 CCK distribution

On receipt of a new SCCK from the SwMI, the ME checks the validity of the CCK-ID then runs the TA32 ALGORITHM to update EF_{CCK}. The record to be updated in EF_{CCK} is identified as follows: The ME checks whether the CCK-ID being broadcast by the SwMI is identical to the CCK-ID stored in record 1 of EF_{CCK}. If not identical, record 1 is updated; otherwise, record 2 is updated.

11.4.2.2 CCK changeover

When the ME detects a new CCK-ID in use it determines the record number in EF_{CCK} which contains the new CCK-ID. After verifying that the new CCK-ID is valid, the ME runs the TA71 ALGORITHM to update all records in EF_{MGCK} using the CCK record in EF_{CCK} identified by the CCK-ID.

11.4.2.3 GCK distribution

The ME analyses EF_{GSSIS} and EF_{GSSID} to locate the required GTSI. If the GTSI is not already present, the ME allocates a free record number in the EF_{GSSID} and there places the new GTSI.

The ME checks whether there is a GCK (and MGCK) associated with the GTSI by accessing the appropriate GCK record number data element in EF_{GRDS} or EF_{GRDD}. If there is no such associated GCK, then a free record in EF_{GCK} is allocated (see note below), and the corresponding target record number in EF_{GRDS} or EF_{GRDD} is updated accordingly.

In the case where there was already a GCK (and MGCK) present, the ME identifies whether the new GCK-VN is valid by comparing it to the GCK-VN being stored currently in the appropriate record of EF_{MGCK}. If it is not valid the procedure is aborted.

The ME then runs the TA82 ALGORITHM to update the respective GCK. After this, the ME runs the TA71 ALGORITHM on this particular GCK to obtain the corresponding MGCK. For this operation, the current CCK (the one being indicated on the broadcast channel) is used.

NOTE: To allocate a free record in EF_{GCK} the ME reads EF_{GRDS} and EF_{GRDD} and works out if there is a record in EF_{GCK} which is not presently pointed to by any GCK record pointer.

11.4.2.4 SCK distribution

On receipt of a new SSCK from the SwMI, the ME identifies whether the new SCK-VN is valid by comparing it to the one being stored currently. If it is not valid the procedure is aborted. Then the ME runs the TA41/52 ALGORITHM in order to unseal the SCK and store it in that record of EF_{SCK}, which is indicated by the SCKN.

11.4.3 ITSI request

The ME performs the reading procedure with EF_{ITSI}.

11.4.4 ITSI disabling/re-enabling

See also EN 300 392-7 [3]

Permanent disabling:

On receiving the ITSI permanent disable command the ME selects EF_{ITSI} and shall then immediately run the SwMI authentication procedure defined in clause 11.4.1.3. If the SwMI is successfully authenticated then the invalidate procedure is performed on $EF_{ITSI.}$ The TETRA session is immediately terminated. (see note)

Temporary disabling:

On receiving the ITSI temporary disable command the ME selects $EF_{ITSIDIS}$ and shall then immediately run the SwMI authentication procedure defined in clause 11.4.1.3. If the SwMI is successfully authenticated then the ME performs the update procedure with $EF_{ITSIDIS}$ to set the flag to "temporarily disabled". (see note)

Re-enabling:

On receiving the ITSI enable command the ME selects $EF_{ITSIDIS}$ and shall then immediately run the SwMI authentication procedure defined in clause 11.4.1.3. If the SwMI is successfully authenticated then the updating procedure is performed on $EF_{ITSIDIS}$ to set the flag to "not disabled".

NOTE: It is an implementation issue for the SIM to deny access to further sensitive EFs (such as group identities and air interface encryption keys) if the ITSI is temporarily or permanently disabled.

11.5 Subscription related procedures

The procedures listed in this clause are only executable if the associated services, which are optional, are provided in the SIM. However, if the procedures are implemented, it shall be in accordance with the requirement stated in this clause. If a procedure is related to a specific service indicated in the SIM service table, it shall only be executed if the corresponding bit denoting this service as "available" (see EF_{SST}). In all other cases this procedure shall not start.

11.5.1 Username request

Requirement: Service no.16 "available".

Request: The ME performs the reading procedure with EF_{UNAME}.

Update: The ME performs the updating procedure with EF_{UNAME}.

11.5.2 ITSI temporarily disabled enquiry

Request: The ME performs the reading procedure with $EF_{ITSIDIS}$.

Update: The ME performs the updating procedure with $EF_{ITSIDIS}$.

11.5.3 Subscriber class request

Request: The ME performs the reading procedure with EF_{SCT}.

Update: The ME performs the updating procedure with EF_{SCT}.

11.5.4 Void

11.5.5 Group identity information

The following procedures apply to both static (EF_{GSSIS}) and dynamic (EF_{GSSID}) groups with the exceptions mentioned in the following clauses.

11.5.5.1 Static Group identity information

Request: The ME performs the reading procedure with EF_{GSSIS}.

11.5.5.2 Dynamic Group identity information

Request: The ME performs the reading procedure with EF_{GSSID}.

Erasure: The ME identifies the record in EF_{GSSID} containing the GSSID to be erased and marks it as free.

Update/invalidate:

The ME selects EF_{GSSID} and shall then immediately run the SwMI authentication procedure defined in clause 11.4.1.3. If the SwMI is successfully authenticated then the update or invalidate procedure is performed on $EF_{GSSID.}$

The update and erasure of EF_{GSSID} requires the updating of the network table. The handling procedures of the network table (EF_{NWT}) are defined under clause 11.6.

11.5.6 Group related data

The following procedures apply to both static and dynamic group related data (EF_{GRDS} and EF_{GRDD}).

Request: The ME performs the reading procedure with EF_{GRDS} or EF_{GRDD}

Update: The ME performs the updating procedure with EF_{GRDS} or EF_{GRDD}

NOTE: A record in EF_{GRDS} or EF_{GRDD} is free when the associated record in EF_{GSSIS} or EF_{GSSID} is marked free.

11.5.7 User's group information

Request: The ME performs the reading procedure with EF_{GINFO}

Update: The ME performs the updating procedure with EF_{GINFO}.

The update of the file is performed in the beginning of a group call.

The update of this file requires the updating of the network table. The handling procedures of the network table (EF_{NWT}) are defined under clause 11.6.

11.5.8 Call modifiers

Requirement: Service no.26 "available".

Request: The ME performs the reading procedure with EF_{CMT}

Update: The ME performs the updating procedure with EF_{CMT}.

11.5.9 Service Provider Name

Requirement: Service no.14 "available".

Request: The ME performs the reading procedure with EF_{SPN}.

11.5.10 DMO channel procedures

Requirement: Service no.27 "available".

Request: The ME performs the reading procedure with EF_{DMOCh}

Update: The ME performs the updating procedure with EF_{DMOCh}.

Erasure: The ME erases the contents of the record in EF_{DMOCh} by filling the record with 'FF'.

11.5.11 Emergency addresses

Request: The ME performs the reading procedure with EF_{EADDR}

Update: The ME performs the updating procedure with EF_{EADDR}.

Erasure: The ME erases the contents of the record in EF_{EADDR} by filling the b1 to b4 in the record with '1'.

11.5.12 Interrupted emergency call request

Request: The ME performs the reading procedure with EF_{EINFO}.

Update: The ME performs the update procedure with EF_{EINFO}

NOTE: If an emergency call was in progress when the ME was powered down the current emergency call record number, if non-zero, indicates that an emergency call procedure was in progress when the ME was powered down. The ME should recognize the non-zero value as an indication to take action as necessary to restart the emergency call after authentication.

11.6 Network related procedures

Request: The ME performs the reading procedure with EF_{NWT}.

Update: The ME checks whether the network address to be stored is already present. If so, the record pointer counter of the found network address record is increased by one.

If the address is not found on the network table, a new record is added to the network table and the corresponding record pointer counter is set to one.

Erasure: The record on the network table is deleted (indicated as free by filling it with 'FF's)

11.6.1 Forbidden networks

Request: The ME performs the reading procedure with EF_{FORBID}.

Update: The ME performs the updating procedure with EF_{FORBID}.

Erasure: The ME can erase the whole contents of the Forbidden networks. The action can either be initiated by the ME or the MMI. In case of erasure, the whole table of Forbidden addresses will be erased i.e. marked free by filling them with 'FF's.

11.6.2 Preferred networks

Requirement: Service no.15 "available".

Request: The ME performs the reading procedure with EF_{PREF}.

Update: The ME performs the updating procedure with EF_{PREF}.

11.7 Dialling number related procedures

The procedures listed in this clause are only executable if the associated services, which are optional, are provided in the SIM. However, if the procedures are implemented, it shall be in accordance with the requirement stated in this clause. If a procedure is related to a specific service indicated in the SIM service table, it shall only be executed if the corresponding bit denoting this service as "available" (see EF_{SST}). In all other cases this procedure shall not start.

11.7.1 Dialling numbers under DF_{TETRA}

The following procedures may be applied to EF_{ADNGWT} and its associated extension file $EF_{GWTEXT1}$ as described in the procedures below. The procedures also refer to EF_{FDNGWT}, EF_{LNDGWT}, EF_{SDNGWT}, $EF_{ADNTETRA}$, $EF_{FDNTETRA}$, $EF_{LNDTETRA}$ and $EF_{SDNTETRA}$ and their associated extension files. If these files are not available, as denoted in the SIM service table, the current procedure shall be aborted and the appropriate EFs shall remain unchanged.

As an example, the following procedures are described as applied to ADNGWT.

Requirement: Service no.3 "available".

Request: The ME sends the identification of the information to be read. The ME shall analyse the data of EF_{ADNGWT} (see clause 10.3.26) to ascertain whether additional data is associated in $EF_{GWTEXT1}$. If necessary, the ME performs the reading procedure on $EF_{GWTEXT1}$ and EF_{GWT} to assemble the complete ADNGWT.

Update: The ME analyses and assembles the information to be stored as follows:

i) the ME identifies the record containing the Name to be updated;

ii) the dialling number (and/or Supplementary service access string in case of ADNTETRA) shall be allocated to the bytes of the EF as follows:

- If the dialling number contains 16 or less "digits", it shall be stored in " number".
- If the dialling number contains more than 16 "digits", the procedure shall be as follows:
- The ME seeks for a free record in $EF_{GWTEXT1}$. If no Extension1 record is marked as "free", the procedure is aborted.
- When a free Gateway Extension 1 record is found, the first 16 "digits" are stored in the " number". The value of the "Length of number contents" is set to the maximum value, which is 16. The Gateway Extension 1 record number in EF_{ADNGWT} is coded with the associated record number in the $EF_{GWTEXT1}$. The remaining digits are stored in the selected Gateway Extension 1 record. The first byte of the Gateway Extension 1 record is set with the number of digits of the remaining data. Further gateway extension records can be added up to the full length of the dialling string by chaining records in Gateway Extension 1. The total number of digits is the sum of the "Length of number contents" of EF_{ADNGWT} and byte 2 of all associated chained Gateway Extension 1 records containing data;

EXAMPLE: A chain of gateway extension records being associated to an ADNGWT or LNDGWT. The Gateway Extension1 record number of ADNGWT or LNDGWT is set to 3.

No of Record	Extension Data	Next	Record
.	.	.	
.	.	.	
Record 3	xx xx	'06'	>——————————┐
Record 4	xx xx	'xx'	│
Record 5	xx xx	'FF'	<————┐ │
Record 6	xx xx	'05'	>————┘ <———┘
.	.	.	
.	.	.	

iii) the ME seeks the gateway address in EF_{GWT}. If it is not already in the table a new entry is created. If a new entry can not be created, the procedure is aborted. When the entry is available the ME updates the Gateway address record number in EF_{ADNGWT} to the associated record in EF_{GWT};

iv) the ME chooses a proper call modifier in EF_{CMT}.

When i), ii), iii) and iv) have been successfully executed the ME performs the updating procedure with EF_{ADNGWT}.

NOTE: If the SIM does not have available empty space to store the received ADN, or if the procedure has been aborted, the ME advises the user.

Erasure: The ME sends the identification of the information to be erased. The content of the identified record in EF_{ADNGWT} is marked as "free". Furthermore, the associated records in EF_{GWT} and $EF_{GWTEXT1}$ are updated accordingly.

11.7.2 Dialling numbers under $DF_{TELECOM}$

The following procedures may be applied to EF_{ADN} and its associated extension file EF_{EXT1} as described in the procedures below, and also to EF_{FDN}, EF_{LND}, EF_{SDN} and their associated extension files. If these files are not available, as denoted in the SIM service table, the current procedure shall be aborted and the appropriate EFs shall remain unchanged.

As an example, the following procedures are described as applied to ADN.

Requirement: Service no. 36 "available".

Request: The ME sends the identification of the information to be read. The ME shall analyse the data of EF_{ADN} (see clause 10.4.1) to ascertain whether additional data is associated in EF_{EXT1}. If necessary, the ME performs the reading procedure on EF_{EXT1} and reading of default gateway SSI from EF_{GWT} to assemble the complete ADN.

Update: The ME analyses and assembles the information to be stored as follows (subscriber has chosen to store ADN to the general EF_{ADN} under $DF_{TELECOM}$):

i) the ME identifies the record containing the Name to be updated;

ii) the dialling number shall be allocated to the bytes of the EF as follows:

- If a “+” is found, the TON identifier is set to “International”
- If the dialling number contains 20 or less "digits", it shall be stored in " Dialling Number/SSC String ".
- If the dialling number contains more than 20 "digits", the procedure shall be as follows:
- The ME seeks for a free record in EF_{EXT1}. If no Extension1 record is marked as "free", the procedure is aborted.
- When a free Extension1 record is found, the first 20 "digits" are stored in the Dialling Number/SSC String. The value of the " Length of BCD number/SSC contents " is set to the maximum value, which is 11. The Extension1 record number in EF_{ADN} is coded with the associated record number in the EF_{EXT1}. The remaining digits are stored in the selected Extension1 record. The first byte of the extension data in EF_{EXT1} (second byte of extension 1 record) is set with the number of digits of the remaining data. Further extension records can be added up to the full length of the dialling string by chaining records in Extension1. The total number of digits is the sum of the " Length of BCD number/SSC contents " of EF_{ADN} and byte 2 of all associated chained extension data records containing data;

iii) if a called party subaddress is associated to the ADN/SSC the procedure shall proceed as follows:

If the length of the called party subaddress is less than or equal to 11 bytes (see GSM 04.08 [14] for coding):

- the ME seeks for a free record in EF EXT1. If an Extension1 record is not marked as "free", the ME runs the Purge procedure. If an Extension1 record is still unavailable, the procedure is aborted;
- the ME stores the called party subaddress in the Extension1 record, and sets the Extension1 record type to "called party subaddress".

If the length of the called party subaddress is greater than 11 bytes (see GSM 04.08 [14] for coding):

- the ME seeks for two free records in EFEXT1. If no such two records are found, the ME runs the Purge procedure. If two Extension1 records are still unavailable, the procedure is aborted;
- the ME stores the called party subaddress in the two Extension1 records. The identifier field in the Extension1 record containing the first part of the subaddress data is coded with the associated EFEXT1 record number containing the second part of the subaddress data. Both Extension1 record types are set to "called party subaddress".

Once i), ii), and iii) have been considered the ME performs the updating procedure with EFADN. If the SIM has no available empty space to store the received ADN/SSC, or if the procedure has been aborted, the ME advises the user.

Erasure: The ME sends the identification of the information to be erased. The content of the identified record in EF_{ADN} is marked as "free". Furthermore, the associated records in EF_{EXT1} are updated accordingly.

Purge: The ME shall access each EF which references EFEXT1 (EFEXT2) for storage and shall identify records in these files using extension data (additional data or called party subaddress). Note that existing chains have to be followed to the end. All referred Extension1 (Extension2) records are noted by the ME. All Extension1 (Extension2) records not noted are then marked by the ME as "free" by setting the whole record to 'FF'.

11.7.3 FDNGWT specific procedures

Requirement: Service no. 5 "available"

If FDN is enabled (i.e. EF_{ADNGWT} is invalidated or not present) the ME shall operate in a restricted mode where only those phone numbers contained in EF_{FDN} and $EF_{FDNGWT}$are used.

If FDNTETRA is enabled (i.e. $EF_{ADNTETRA}$ is invalidated or not present) the ME shall operate in a restricted mode where only those phone numbers contained in $EF_{FDNTETRA}$ are used.

Both modes FDN and FDNTETRA can be enabled independently of each other.

ADNGWT and FDNGWT are mutually exclusive of each other and independent of the state of ADNTETRA and FDNTETRA. Likewise, ADNTETRA and FDNTETRA are mutually exclusive of each other and independent of the state of ADNGWT and FDNGWT. This means that there may be restricted ADNGWT phonebook operation or restricted TETRA phonebook operation and these are independent of each other.

The following three procedures are only applicable to service no.4 (FDNTETRA) no. 5 (FDNGWT). As an example, the following procedures are described as applied to FDNGWT.

11.7.3.1 FDNGWT capability request

To ascertain the state of FDNGWT, the ME checks in EF_{SST} whether or not ADNGWT is activated. If ADNGWT is not activated, service no. 5 is enabled. If ADNGWT is activated, the ME checks the response data EF_{ADNGWT}. If EF_{ADNGWT} are invalidated, service no. 5 is enabled. In all other cases service no. 5 is disabled.

11.7.3.2 FDNGWT disabling

The FDNGWT disabling procedure requires that CHV2 verification procedure has been performed successfully and that ADNGWT is activated. If not, FDNGWT disabling procedure will not be executed successfully. To disable FDNGWT capability, the ME rehabilitates EF_{ADNGWT}. The invalidate/rehabilitate flag of EF_{ADNGWT}, which are set by the REHABILITATE command, is at the same time the indicator for the state of the service no. 5. If ADNGWT is not activated, disabling of FDNGWT is not possible and thus service no. 5 is always enabled (see FDNGWT capability request).

11.7.3.3 FDNGWT enabling

The FDNGWT enabling procedure requires that CHV2 verification procedure has been performed successfully. If not, FDNGWT enabling procedure will not be executed successfully. To enable FDNGWT capability, the ME invalidates EF_{ADNGWT}. The invalidate/rehabilitate flag of EF_{ADNGWT}, which is set by the INVALIDATE command, is at the same time the indicator for the state of the service no. 5 (see FDNGWT capability request). If ADNGWT is not activated, service no. 5 is always enabled.

Invalidated ADNGWTs may optionally still be readable and updateable depending on the file status (see clause 9.4).

11.8 Status and short data message procedures

The procedures listed in this clause are only executable if the associated services, which are optional, are provided in the SIM. However, if the procedures are implemented, it shall be in accordance with the requirement stated in this clause. If a procedure is related to a specific service indicated in the SIM service table, it shall only be executed if the corresponding bit denoting this service as "available" (see EF_{SST}). In all other cases this procedure shall not start.

11.8.1 Display of status message texts

Requirement: Service no.22 "available".

Request: The SIM selects EF_{STXT} and searches for the identified status message value. If the message value is found it performs the reading procedure with EF_{STXT}.

11.8.2 Display of SDS1 message texts

Requirement: Service no.23 "available".

Request: The SIM selects EF_{MSGTXT} and searches for the identified status message value. If the message value is found it performs the reading procedure with EF_{MSGTXT}.

11.8.3 Storage of status and SDS messages types 1, 2 and 3

Requirement: Service no.24 "available".

Request: The SIM selects EF_{SDS123} and searches for the identified status or SDS message. If this message is found, the ME performs the reading procedure with EF_{SDS123}

Update: The ME looks for the next available area to store the status or SDS message in EF_{SDS123}. If such an area is available, it performs the updating procedure with $EF_{SDS1123.}$

If there is no available empty space in the SIM to store the received short message, the ME advises the user.

Erasure: The ME selects EF_{SDS123} and identifies the records to be erased. Then it performs the update procedure to mark them as free.

NOTE: Depending on the ME, the message may be read before the record is marked as "free". After performing the updating procedure with EF_{SDS123}, the memory allocated to this short message in the SIM is made available for a new incoming message. The memory of the SIM may still contain the old message until a new message is stored in that area.

11.8.4 Storage of SDS messages type 4

Requirement: Service no.25 "available".

Request: The SIM selects EF_{SDS4} and searches for the identified short message. If this message is found, the ME performs the reading procedure.

Update: The ME looks for the next available area to store the short message in EF_{SDS4}. If such an area is available, it performs the updating procedure with EF_{SDS4}.

If there is no available empty space in the SIM to store the received short message, the ME advises the user

Erasure: The ME selects EF_{SDS4} and identifies the records to be erased. Then it performs the update procedure to mark them as free.

NOTE: Depending on the ME, the message may be read before the record is marked as "free". After performing the updating procedure with EF_{SDS123}, the memory allocated to this short message in the SIM is made available for a new incoming message. The memory of the SIM may still contain the old message until a new message is stored in that area.

11.8.5 SDS delivery report

Requirement: Service number 32 "available".

Request: If the status of a stored short message indicates that there is a corresponding status report, the ME performs the seek function with EF_{SDSR} to identify the record containing the appropriate status report. The ME performs the reading procedure with EF_{SDSR}.

Update: If the status report is received, the ME first seeks within the SDS record identifiers of EF_{SDSR} for the same record number it used for the short message in EF_{SDS4}. If such a record identifier is found in EF_{SDSR}, it is used for storage. If such a record identifier is not found, then the ME seeks for a free entry in EF_{SDSR} for storage. If no free entry is found, the ME runs the Purge procedure with EF_{SDSR}. If there is still no free entry, the status report is not stored.

If the ME found an appropriate record in EF_{SDSR} for storage, it updates the record with the status report setting the record identifier in EF_{SDSR} to the appropriate record number of the short message in EF_{SDS4}.

The status in EF_{SDS4} is updated accordingly (see sub clause 10.3.42) by performing update procedure with EF_{SDS4}.

Erasure: The ME runs the update procedure with EF_{SDSR} by storing '00' in the first byte of the record.

Purge: The ME shall read the SDS record identifier (byte 1) of each record of EF_{SDSR}. With each record the ME checks the corresponding SDS message in EF_{SDS4}. If the status of the corresponding SDS is not equal to 'status report requested, received and stored in EF_{SDSR}' the ME shall perform the erasure procedure with the appropriate record in EF_{SDSR}.

11.8.6 Default Status Target

Requirement: Service number 31 "available".

Request: The ME checks whether a destination address has been specified if not then the ME performs the read procedure with $EF_{DFLTSTSTGT}$.

Update: The ME runs the update procedure with $EF_{DFLTSTSTGT}$.

Annex A (normative): Plug-in SIM

This annex specifies the dimensions of the Plug-in SIM as well as the dimensions and location of the contacts of the Plug-in SIM. For further details of the Plug-in SIM see clause 4.

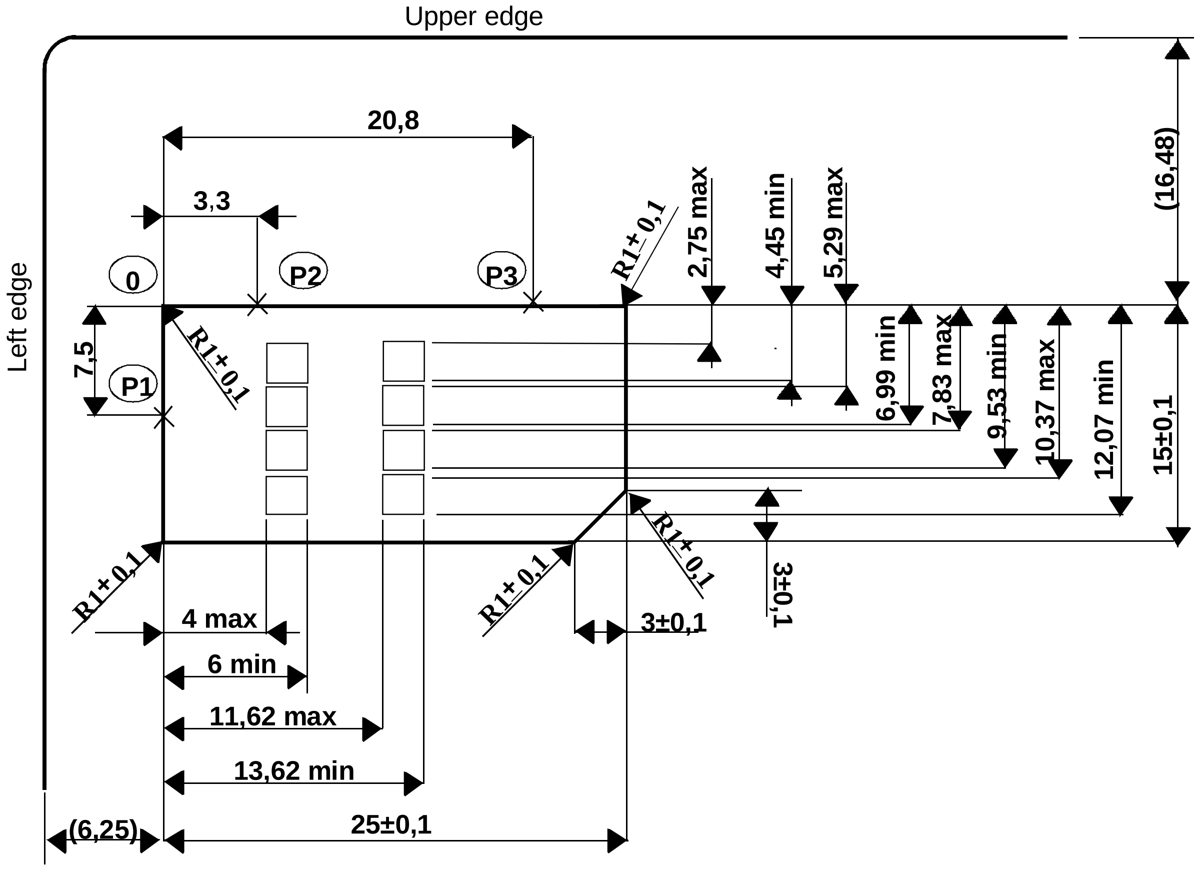


Figure A.1: Plug-in SIM

NOTE: The Plug-in SIM may be "obtained" by cutting away excessive plastic of an ID-1 SIM. The values in parenthesis in figure A.1 show the positional relationship between the Plug-in and the ID-1 SIM and are for information only.

Annex B (informative): FDN Procedures

The FDN facility allows operation of the TETRA terminal in a restricted state whereby it can only initiate calls to a pre-determined destination or list of destinations.

A TETRA SIM may be personalized so that the terminal can be operated in only the restricted state, only the unrestricted state or to allow the operation mode to be switched between states through the MMI.

FDN services

Two FDN services are provided for the TETRA SIM. Service number 4 allows fixed dialling to other TETRA addresses while service number 5 allows fixed dialling to destinations on a PABX or the PSTN. These services may be individually or jointly enabled as indicated in the SIM service table.

The SIM service table provides an enable/disable indicator for each of the two FDN services to indicate to the ME the capabilities of the SIM. Where the SIM service table indicates that the SIM is capable of both ADN and FDN services, the operating state can be switched as described below.

FDN operation

When the ME is operating in the restricted FDN state, the user may only call destinations listed in the FDN directories EF_{FDN} (service no 5) and/or $EF_{FDNTETRA}$ (service no. 4). Attempts to call other destinations shall be rejected by the ME, other than those initiated by activation of the emergency call procedures.

FDN initialization

When a TETRA session is initialized, the ME should check the SIM service table for the state of the FDN services. If neither service is enabled, the ME should enter the unrestricted operation state, offering facilities as otherwise indicated in the SIM service table.

If either of the FDN services are enabled in the SIM service table, the ME should further check the entries for ADN (service no. 2) and ADNTETRA (service no. 3). If neither ADN service is enabled the ME should enter the restricted FDN operation state.

If both ADN and FDN services are enabled in the SIM service table, the operation mode may be determined by the validity of EF_{ADN}. If EF_{ADN} is invalidated, the ME should enter the restricted FDN operation state. If EF_{ADN} is not invalidated, the ME should enter the unrestricted state.

Change of FDN operation mode.

Where the SIM Service Table indicates that a SIM supports both FDN and unrestricted modes of operation, the validity of the file EF_{ADN} provides the indicator as to the current operating state as described above.

The ME may provide an MMI operation to allow toggling of the operation state by performing invalidation or rehabilitation of EF_{ADN}. This procedure can only be performed after successful completion of the CHV2 verification procedure to satisfy the access rights for EF_{ADN}.

Change of FDN access details

The ME may provide a method on the MMI to change entries in the FDN directories, thereby changing the list of call destination when the ME is operating in the restricted state. This procedure can only be performed after successful completion of the CHV2 verification procedure to satisfy the access rights for update to EF_{FDN}.

Annex C (informative): Suggested contents of EFs at pre-personalization

If EFs have an unassigned value, it may not be clear from the main text what this value should be after conclusion of the manufacturing phase and prior to personalization of initial usage. This annex suggests values in these cases.

The values stored in EF_{CCK}, EF_{SCK}, EF_{GCK} and EF_{MGCK} may only be changed using the appropriate OTAR algorithms in the TAA1 set. The initial values to be stored may be assigned by the network operator and loaded during the manufacturing phase. If particular values are not assigned it is suggested that these files are populated with a null value, '00 … 00'.

C.1 Contents of the EFs at the MF level

File Identification	Description	Value
EF_{ICCID}	Card identification	Operator dependent (see clause 10.2.1)
EF_{DIR}	Application directory	'FF…FF'
EF_{LP}	Language preference	Operator dependent (see clause 10.2.3)

C.2 Contents of the EFs at the TETRA application level

File Identification	Description	Value
EF_{SST}	SIM Service Table	Operator dependent (see clause 10.3.1), else '00…00'
EF_{ITSI}	ITSI	Operator dependent (see clause 10.3.2)
$EF_{ITSIDIS}$	ITSI Disabled	'00'
EF_{UNAME}	Username	'FF…FF'
EF_{SCT}	Subscriber class table	Operator dependent (see clause 10.3.5)
EF_{PHASE}	Phase identification	'01'
EF_{CCK}	Common Cipher Key	Operator dependent (see clause 10.3.7)
EF_{CCKLOC}	CCK Location Areas	Operator dependent (see clause 10.3.8)
EF_{SCK}	Static Cipher Key	Operator dependent (see clause 10.3.9)
EF_{GSSIS}	Pre-programmed GSSIs	Operator dependent (see clause 10.2.10)
EF_{GRDS}	Group related data for Static GSSIs	Operator dependent (see clause 10.3.11), else 'FF…FF'
EF_{GSSID}	Dynamic GSSIs	'FF…FF'
EF_{GRDD}	Group related data for Dynamic GSSIs	'FF…FF'
EF_{GCK}	Group Cipher Keys	Operator dependent (see clause 10.2.14)
EF_{MGCK}	Modified Group Cipher Keys	Operator dependent (see sub-clause 10.3.15)
EF_{GINFO}	User's group information	Operator dependent (see clause 10.3.16), else '00 00FF...FF FF 00 FF FF FF'
EF_{FORBID}	Forbidden networks table	Operator dependent (see clause 10.3.18), else 'FF...FF'
EF_{PREF}	Preferred networks table	Operator dependent (see clause 10.3.19), else 'FF...FF'
EF_{SPN}	Service Provider Name	'FF...FF
EF_{DNWRK}	Broadcast network information	'00…00'
EF_{NWT}	Network table	1st record operator dependent (see clause 10.3.24), else 'FF…FF'
EF_{GWT}	Gateway Table	Operator dependent (see clause 10.3.24), else 'FF…FF'
EF_{CMT}	Call modifier table	'FF...FF'

File Identification	Description	Value
EF_{ADNGWT}	Abbreviated Dialling Number with Gateway	'FF...FF'
$EF_{GWTEXT1}$	Gateway Extension 1	'FF...FF'
$EF_{ADNTETRA}$	Abbreviated Dialling Numbers for TETRA network	'FF...FF'
EF_{EXTA}	Extension A	'FF...FF'
EF_{FDNGWT}	Fixed Dialling Number with Gateway	'FF...FF'
$EF_{GWTEXT2}$	Gateway Extension 2	'FF...FF'
$EF_{FDNTETRA}$	Fixed Dialling Numbers for TETRA network	'FF...FF'
EF_{LNDGWT}	Last Number Dialled with Gateway	'FF...FF'
$EF_{LNDTETRA}$	Last Number Dialled for TETRA network	'FF...FF'
EF_{SDNGWT}	Service Dialling Number with Gateway	'FF...FF'
$EF_{GWTEXT3}$	Gateway Extension 3	'FF...FF'
$EF_{SDNTETRA}$	Service Dialling Numbers for TETRA network	'FF...FF'
EF_{STXT}	Status message texts	Operator dependent (see clause 10.3.39)
EF_{MSGTXT}	SDS-1 message texts	'FF...FF'
EF_{SDS123}	Status and SDS type 1, 2 and 3 message storage	'FF...FF'
EF_{SDS4}	SDS type 4 message storage	'FF...FF'
EF_{MSGEXT}	Message Extension	'FF...FF'
EF_{EADDR}	Emergency address	'FF...FF'
EF_{EINFO}	Emergency call information	'00'
EF_{DMOCh}	DMO Channel Information	'FF...FF'
EF_{MSCh}	MS allocation of DMO channels	'FF...FF'
EF_{KH}	List of Key Holders	See clause 10.3.48
$EF_{REPGATE}$	DMO repeater and gateway list	'FF...FF'
EF_{AD}	Administrative Data	See clause 10.3.50
EF_{PREF_LA}	Preferred Location Areas	'FF...FF'
$EF_{LNDComp}$	Composite LND file	'FF...FF'
$EF_{DFLTSTSTGT}$	Default Status Target	'FF...FF'
EF_{SDSMEM_STATUS}	SDS Memory Status	'FF...FF'
$EF_{WELCOME}$	Welcome message	Operator dependent (see clause 10.3.55), else 'FF...FF'
EF_{SDSR}	SDS delivery report	'00...00'
EF_{SDSP}	SDS Parameters	'FF...FF'
EF_{DIALSC}	Dialling schemes for TETRA network	'FF...FF'
EF_{PNI}	Private Number Information	'FF…FF'

C.3 Contents of the EFs at the Telecom Level

File Identification	Description	Value
EF_{ADN}	Abbreviated Dialling Numbers	'FF...FF'
EF_{FDN}	Fixed Dialling Numbers	'FF...FF'
EF_{MSISDN}	MSISDN	'FF...FF'
EF_{LND}	Last Number Dialled	'FF...FF'
EF_{SDN}	Service Dialling Numbers	'FF...FF'
EF_{EXT1}	Extension 1	'FF...FF'
EF_{EXT2}	Extension 2	'FF...FF'
EF_{EXT3}	Extension 3	'FF...FF'

Annex D (normative): Database structure for group IDs and phone books

Use of the network table

Relational database mechanisms are used to save a significant amount of memory. Several EFs (e.g. EF$_{GSSIS}$ and EF$_{GSSID}$) refer to the Network table for network address instead of saving it with each group short subscriber identity. However, since a network address can be referenced from more than one place, a record pointer counter is needed to keep track of how many times a network address is referenced. When the record pointer counter of a network address is one, it is referenced from only one place. When that address is removed, the corresponding network address can be removed also, since it was the only one using it. This housekeeping method is used to remove unnecessary network addresses from the network table.

The network table is thus handled using the following procedures:

- When a network address needs to be stored with a record, the network table (EF$_{NWT}$ see clause 10.3.23) needs to be read. If the address (MCC and MNC) is already found on the network table, the Record pointer counter of the found network address record needs to be increased by one. Only the record number of the network address on the network table is stored with the record that needs the network address.

- If the address is not found on the network table, a new record needs to be added to the network table. On the network table the new network address (MCC and MNC) is stored along with a record pointer counter, which is set to one. Only the record number of the network address on the network table is stored with the record that needs the network address.

- If the desired network address is not found in the network table, and it cannot be added because of the file being full, the new network address cannot be stored on the SIM.

- If a record that uses a network address in the network table needs to be deleted, the network table also needs to be updated. The record that needs to be updated can be found using the record number. The record number is stored with the record that is to be deleted. When the record in the network table is found, the record pointer counter is read. If the value of the counter is 2 or higher, the counter is decreased by one and the record that referenced it can be deleted.

- If the record pointer counter is 1, the whole record on the network table can be deleted (indicated as free by filling it with 'FF's) along with the record that pointed to that record.

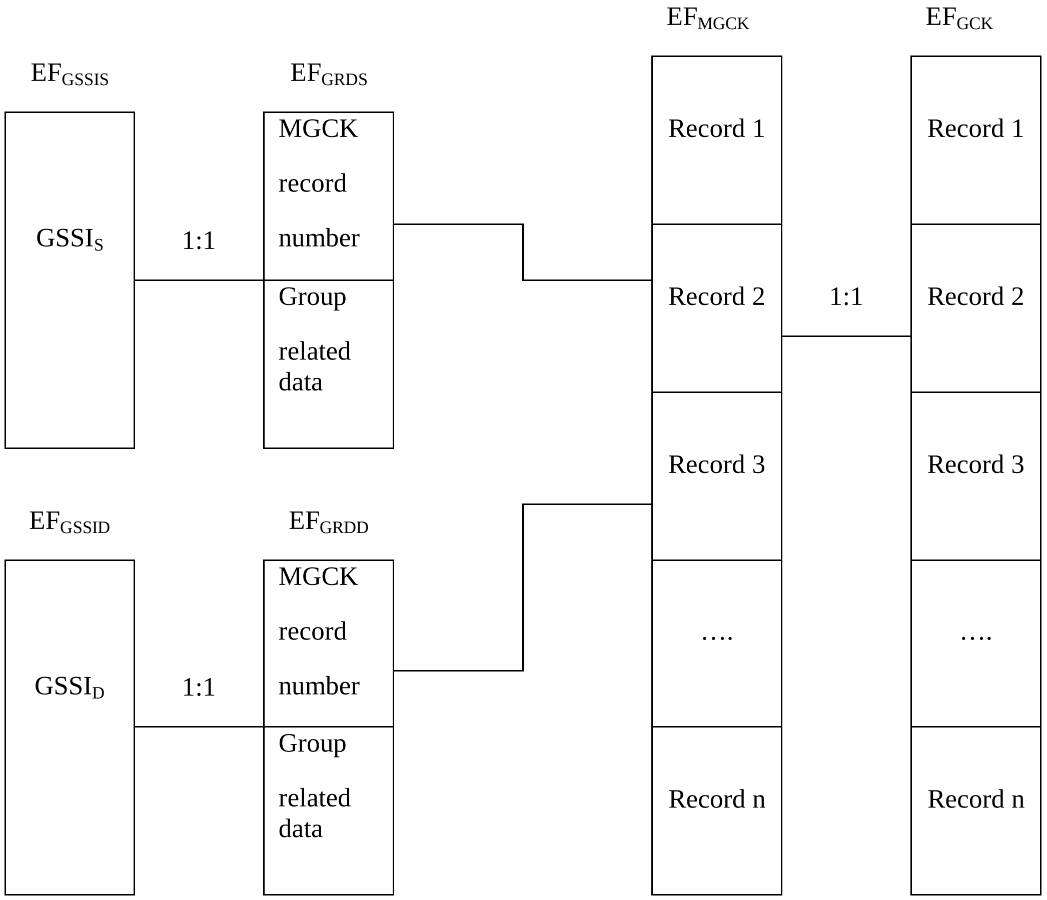


Figure D.1: Graphical presentation of group data related EF structures

Figure D.2 shows how records in phonebook related EFs can point to records in other phonebook related EFs.

NOTE: Each of the 8 phonebooks (ADNGWT, LNDGWT, FDNGWT, SDNGWT, ADNTETRA, LNDTETRA, FDNTETRA and SDNTETRA) may point to EF_{CMT}, which is not shown on the diagram.

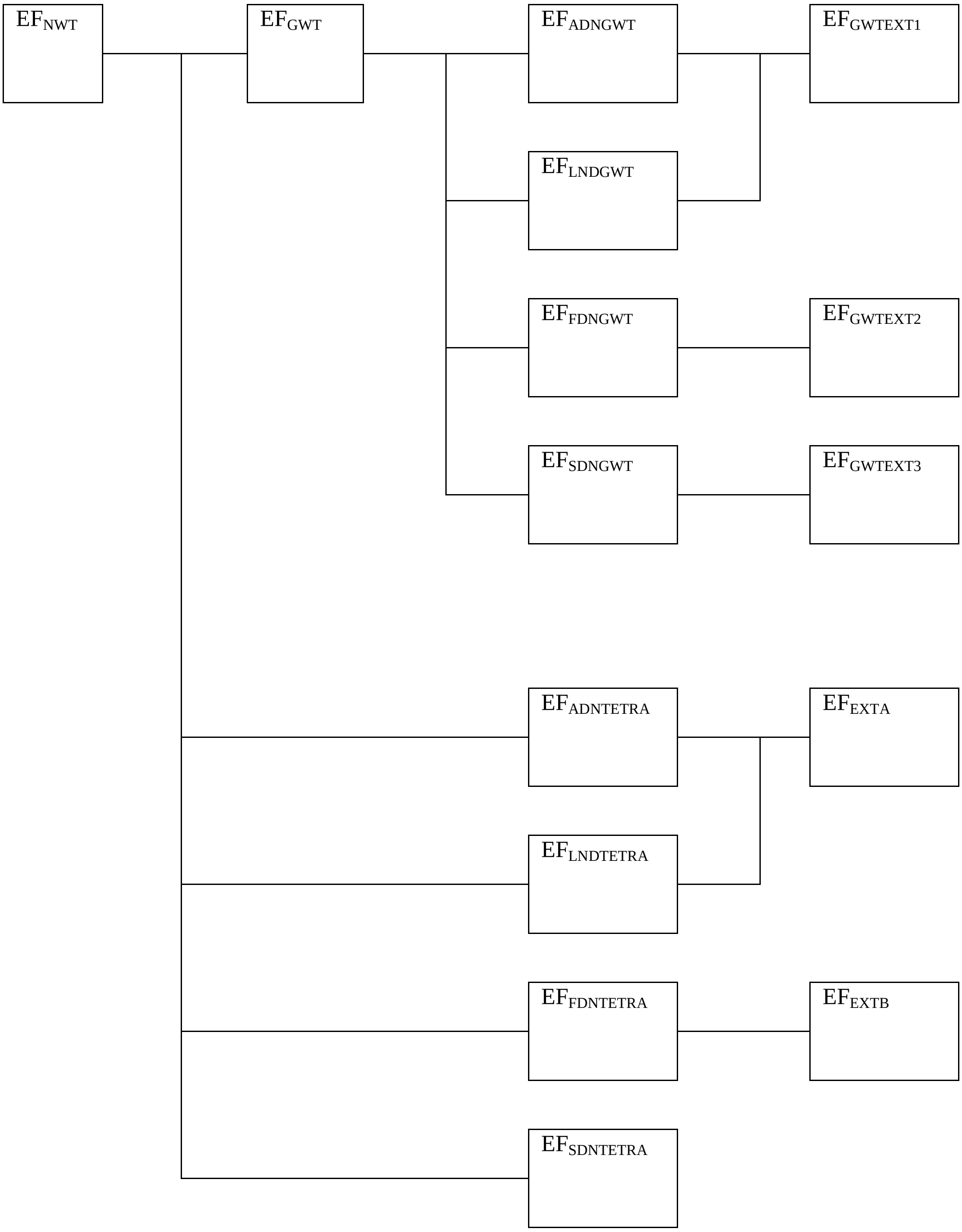


Figure D.2: Graphical presentation of phonebook related EF structures

Annex E (informative): Emergency call facilities and procedures.

The TETRA standards provide a wide variety of call types and facilities which may be used in an emergency situation. The activation of an emergency facility is implementation-specific and so the file content defined for the TETRA SIM card is intended to offer flexibility in handling emergency situations. This annex offers further explanation of the information available to the ME in handling an emergency situation.

Emergency call control

The EF_{EINFO} contains a control flag to indicate to the whether or not emergency calls are enabled for this particular card.

Emergency call addresses

The EF_{EADDR} contains a list of call destinations for use in an emergency call. Entries in the file can require that the call be placed to either the last group in which the ME took part or to a pre-defined destination. When the file contains more than one address, it is suggested that the order of the records in the file should indicate the order of preference for the call, starting with the highest preference.

Each record in EF_{EADDR} also contains a number of flags providing an indication as to the type of the call address, allowing a mix of call types to be indicated. The call type can be one of a selection of 10 variants, including all of the common speech calls and short data transactions. For circuit mode calls, a data field indicates the nature of the required call i.e. individual, group, acknowledged group or broadcast.

When the emergency call type is a status or short data transaction, an additional option is selected by a flag which may be used to indicate a preference as to the source of the data to be transferred in an emergency message. When the pre-defined value stored in the card is selected, a record number pointer indicates EF_{SDS123} or EF_{SDS4} which contain both the destination and message content. When the "application" source is selected, it is suggested that the contents of the data field would be obtained by an application running in the ME.

Protection for interrupted emergency calls

The EF EF_{EINFO} contains a flag indicating the action to be taken on power-on after an interrupted emergency call - to optionally resume the emergency call without further operator intervention.

Where EF_{EINFO} indicates that an interrupted emergency call should be continued next time the ME is powered up, the ME should maintain the current emergency call index in EF_{EINFO} during any emergency call procedure. In particular, the index should be set by the ME to a value to be understood by the restarting ME as the call is initiated and zeroed on normal termination. The index allows the restarting ME to establish that an emergency transaction was in progress and, from the index, which of the available call options to restart. The coding of the index is implementation-dependant but is dimensioned so that it can be used as a pointer to a record number in EF_{EADDR} if required.

Successful connection of an emergency call

It is suggested above that the ME should attempt to set up the emergency call to each of the destinations prescribed in EF_{EADDR} until a successful connection is achieved.

It should, however, be noted that not all call types provide a definite indication of success. An unacknowledged group call, for example, may succeed in establishing a 'call' but it is possible that no other member of the group could be available and so the result would be no exchange of useful information. For PABX or PSTN voice calls, call routing beyond the TETRA infrastructure may not be able to return a definite indication of a successful exchange to the originating terminal and so a call to an unanswered or engaged number could result. The implementation of the emergency facility may take account of this possibility in controlling the emergency call.

Emergency calls in Direct Mode

When an emergency call record in EF_{EADDR} requires the use of direct mode, the implementation may handle the possibility of the required party being on one of a multiplicity of DMO channels. The record in EF_{EADDR} includes a field to indicate a channel number explicitly. It is suggested that a zero channel number could cause the ME to use the flags provided in EF_{DMOCh} which designate a channel for emergency use in attempting to set up the call.

Emergency calls when the SIM card is not fitted

Where the ME is not equipped with a SIM interface, or the SIM is absent, it must still be possible, for some applications, to make an emergency call.

Annex F (informative): Composite List of Last Dialled Numbers.

Each phonebook has a distinct file holding a list of Last Numbers Dialled (LND). When a subscriber initiates a call in a particular mode, the called number is written to the corresponding LND file. Table F 1 summarizes the link between the handset mode, phonebook elementary file and the LND elementary file.

Table F.1

Mode	Phonebook	Last Number Dialled
PSTN	EF_{ADN}	EF_{LND}
PABX	EF_{ADNGWT}	EF_{LNDGWT}
PRIVATE	$EF_{ADNTETRA}$	$EF_{LNDTETRA}$
GROUP	EF_{GSSIS}/EF_{GSSID}	Non-existent

The navigation of the MMI may be simplified for the user if only one (composite) list of Last Dialled Numbers is maintained to permit the user to review the Last Numbers Dialled in reverse chronological order. The composite LND file enables this functionality to be offered because each mode (except GROUP) has a distinct LND file and entries in these files are not timestamped and therefore cannot be sorted in time.

Operation of $EF_{LNDComp}$

The composite LND file is updated with a pointer to the relevant individual LND file when a call is originated. The pointer includes the file identifier and record number for the relevant LND file. The relationship between the files is shown in Figure F 1.

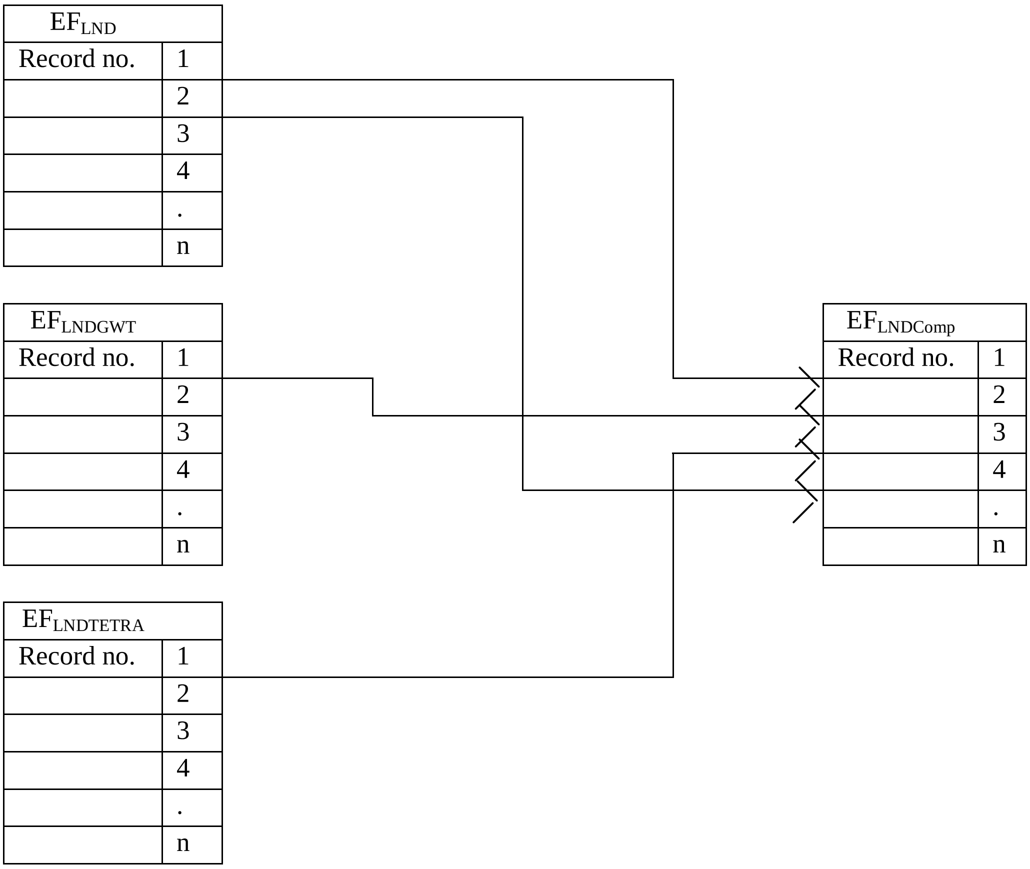


Figure F.1: Graphical representation of relationship between the LND files

It is recommended that a maximum file length equal to the length of one of the individual LND files is used. The reasoning is that if $EF_{LNDComp}$ is longer than one of the individual LND files it will be quicker to find the original dialling number in the phone books.

Annex G (informative): Bibliography

- ETSI EN 726-3: "Terminal Equipment (TE); Requirements for Integrated circuit (IC) cards and terminals for telecommunication use: Part 3: Application independent card requirements".

- ETSI EN 726-4: "Terminal Equipment (TE); Requirements for Integrated circuit (IC) cards and terminals for telecommunication use: Part 4: Application independent card related terminal requirements".

History

Document history		
V2.1.1	August 2001	One-step Approval Procedure OAP 20011214: 2001-08-15 to 2001-12-14