3GPP2 C.P0065-B v2.15

November 2013

cdma2000 Application on UICC for Spread Spectrum Systems

Revision History

Revision	**Description**	**Date**
C.S0065-0 v1.0	Initial Release	June 2006
C.S0065-0 v2.0	Point Release	July 2008
	Corrected Point Release	December 2008
C.S0065-A v1.0	Revision A	August 2009
C.S0065-B v1.0	Revision B	January 2010
C.S0065-B v2.0	Revision B Version 2.0	January 2011
C.S0065-B v3.0	Revision B v3.0	xxxx 2013

INDEX OF CHANGES FOR C.S0065-B V3.0

Section	Name	Changes	Source	v
Cover	Cover	• Remove word "Date" before publication date.	C.S0023-Dv2.0	2.01
1.1	Scope	• Demoted to level 2 header	C.S0023-Dv2.0	2.01
1.2	Requirements Language	• Added this new section	C.S0023-Dv2.0	2.01
1.1	Scope	• Renumber as Chapter 2 to keep numbering aligned with previous versions.	Editor	2.04
1.3.1	Normative References	• Number this section	C.S0023-Dv2.0	2.01
		• Delete unused references 2, 4, 16, 20, 24, 25, 26, 27, 29, 31, 32, 33, 34, 42, 49, 52	C.S0023-Dv2.0 Editor	2.01
		• Update several references to latest version	C.S0023-Dv2.0 Editor	2.01
		• Add RFCs from EF-AuthCapability as [69] through [74]	C.S0023-Dv2.0 Editor	2.01
		• Add publication date for C.s0016-D v2.0 and consequently remove editor's note.	Editor	2.04
		• Add RFC 2002	20130909-015r1	2.14
1.3.2	Informative References	• Number this section	C.S0023-Dv2.0	2.01
4	Definitions	• Clarify terms related to UIMID and MEID and make spelling consistent.	C.S0023-Dv2.0	2.01
		• Change pseudo-identifier acronyms to pESN and pUIMID	C.S0023-Dv2.0	2.01
		• Add ECC definition.	20130311-010R3	2.08
		• Change references for AID, PIX, RID.	20130603-012	2.11
4.2	Parameters Stored Temporarily in the CSIM	• Add this new section	C.S0023-Dv2.0	2.01
5	Coding of EFs…	• Prefix sentence with word "References" • Bulletize paragraph describing types of variables. • Add description of temporary variables.	C.S0023-Dv2.0	2.01
		• Indent bullets in line with similar paragraphs	C.S0023-Dv2.0	2.01
5.1	Contents of files at the MF level	• Clarify locations of the definition of AID, RID and PIX. • Clarify how multiple CSIM applications are distinguished.	20130603-012	2.11
5.2.2	EF_{IMSI_M}	• Be more specific about location of bits in multi-octet integers	C.S0023-Dv2.0	2.01
		• Encoding order. • How IMSI_M_PROGRAMMED is set • Spelling of IMSI_M_ADDR_NUM	C.S0023-Dv2.0	2.01
		Add reference to parallel section of [5] (C.S0005) as well as section reference to [14] (TIA-95-B).	C.S0023-Dv2.0	2.01

Section	Name	Changes	Source	v
5.2.3	EF_{IMSI_T}	• Add reference to [5] (C.S0005) as well as [14] (TIA-95-B)	C.S0023-Dv2.0	2.01
5.2.4	EF_{TMSI} (TMSI)	• Add reference to parallel section in [5].	C.S0023-Dv2.0	2.01
		• Remove reference to unnecessary sub-sections	C.S0023-Dv2.0	2.01
5.2.5	EF_{AH}	• Make analog EF optional	C.S0023-Dv2.0	2.01
5.2.6	EF_{AOP}	• Make analog EF optional	C.S0023-Dv2.0	2.01
5.2.7	EF_{ALOC}	• Make analog EF optional	C.S0023-Dv2.0	2.01
5.2.9	EF_{ZNREGI}	• Change [15/14] to "[5] and [14]"	C.S0023-Dv2.0	2.01
5.2.10	EF_{SNREGI}	• Add reference to [5] in addition to [14].	C.S0023-Dv2.0	2.01
5.2.11	$EF_{DISTREGI}$	• Add reference to [5] in addition to [14].	C.S0023-Dv2.0	2.01
5.2.14	EF_{SSCI}	• Add reference to [5] in addition to [14].	C.S0023-Dv2.0	2.01
5.2.15	EF_{ACP}	• Make analog EF optional	C.S0023-Dv2.0	2.01
5.2.16	EF_{PRL}	• Correct references	C.S0023-Dv2.0	2.01
5.2.18	EF_{CSIM_ST}	• Clarification of FDN service	120611-022r1	2.03
		• Consolidate sections describing bits for available/not available. • Remove ° character	Editor	2.03
		• Clarify references to the three invalidated EFs with FDN	C.S0023-Dv2.0	2.01
		• Clarifications related to deprecating service n32. • Simplification of CSIM_ST/EST interactions	20121210-025r1	2.06
		• Add new call control services for SMS and Data • Clarify that existing "Call Control" service is just for voice.	20130311-035	2.08
		• Reserve service n33 (CCP) • Add footnote indicating that this service has been deprecated. • Update previous footnote to also indicate name of deprecated service.	20130514-006	2.10
		• Add service n44 for support of USSD ENVELOPE commands	20131007-014	2.15
5.2.20	$EF_{OTAPASPC}$	• Align spec with OTASP to allow user to deny change only from default to non-default SPC value.	C.S0023-Dv2.0	2.01
5.2.21	$EF_{NAMLOCK}$	• Capitalize "OTAPA REQUEST"	20121015-008	2.05
		• Clarify which bits can be changed by ME and which cannot. • Clarify relationship of bits to OTASP	20120806-005	2.04
5.2.23	EFSP (Service Preferences)	• Add reference to 2.3.10.1 of [5] (but not 2.3.10.2)	C.S0023-Dv2.0	2.01
5.2.26	EF_{LI}	• Clarify character encoding and language indicator.	20100520-016	2.01
		• Replace "specified" relative to an informative in two places	20100826-009r1	2.01

Section	Name	Changes	Source	v
5.2.27	EF_{FDN}	• Mark byte X+13 (EF_{CCP2} Record Identifier) as reserved ('FF').	20130514-006	2.10
5.2.29	EF_{SMSP}	• Clarification of encoding (within table).	C.S0023-Dv2.0	2.01
5.2.31	EF_{SSFC}	• Corrections to grammar. • Numbers are not "coded as hexadecimal", that is a display format.	Editor	2.03
		• Digits are not right-justified. Encoding shows that they are left justified. Remove misleading text.	20120718-015	2.03
5.2.32	EF_{SPN}	• Note that this EF is needed if service n10 is allocated. • Clarify that b1='1' means required and '0' means not required.	C.S0023-Dv2.0	2.01
		• Clarify that this is home service provider name, not registered system name.	20120516-009	2.02
		• Clarify encoding with 7 bit characters	20130711-005	2.12
5.2.33	EF_{USGIND}	• b2 is conditional on service n34 (C.S0023 n8)	C.S0023-Dv2.0 and Editor	2.01
		• Remove first two lines of coding description, in favor of the clearer text in the Byte 1 coding description.	20130709-004	2.12
5.2.34	EF_{AD}	• Fix incorrect curly quote	C.S0023-Dv2.0	2.01
5.2.35	EF_{MDN}	• Explain bit layout of "Number of Digits" field outside the table.	C.S0023-Dv2.0	2.01
		• Explain encoding of last octet of MDN field.	C.S0023-Dv2.0	2.01
		• Add references from [5] parallel to those from [14]	C.S0023-Dv2.0	2.01
5.2.36	EF_{EPRL}	• Change "EPRL" to subscript (two times)	Editor	2.15
5.2.37	EF_{SPCS}	• Change "NEVER" to "Never" in 3 places	Editor	2.09
5.2.38	EF_{ECC}	• Add acronym	20130311-010R3	2.08
		• Remove restriction to 5 codes	20130311-028	2.09
5.2.39	$EF_{ME3GPDOPC}$	• Change "SimpleIP" to "Simple IP" and "MobileIP" to "Mobile IP"	C.S0023-Dv2.0	2.01
5.2.41	EF_{SIPCAP}	• Change "SimpleIP" to "Simple IP" in title (leave references to C.S0016 alone)	C.S0023-Dv2.0	2.01
5.2.42	EF_{MIPCAP}	• Change "MobileIP" to "Mobile IP" in title (leave references to C.S0016 alone)	C.S0023-Dv2.0	2.01
5.2.43	EF_{SIPUPP}	• Change "SimpleIP" to "Simple IP" in title (leave references to C.S0016 alone)	C.S0023-Dv2.0	2.01
5.2.44	EF_{MIPUPP}	• Change "MobileIP" to "Mobile IP" in title (leave references to C.S0016 alone)	20110609-014	2.01
5.2.45	EF_{SIPSP}	• Change "SimpleIP" to "Simple IP" in title (leave references to C.S0016 alone)	C.S0023-Dv2.0	2.01
5.2.46	EF_{MIPSP}	• Change "MobileIP" to "Mobile IP" in title (leave references to C.S0016 alone)	C.S0023-Dv2.0	2.01
5.2.47	$EF_{SIPPAPSS}$	• Change "SimpleIP" to "Simple IP" in title (leave references to C.S0016 alone)	C.S0023-Dv2.0	2.01
5.2.52	EF_{MECRP}	• Reference C.S0005 [5] for source of terms SCM and "local control" for CDMA • Expand term SCM	C.S0023-Dv2.0	2.01

Section	Name	Changes	Source	v
		• Align Local Control terminology with OTASP spec	C.S0023-Dv2.0	2.01
		• Capitalize name of command	20121015-008	2.05
5.2.55	EF_{CSSPR}	• Clarify that this EF is not really needed, and only applicable to EF_{EPRL} not EF_{PRL}.	C.S0023-Dv2.0	2.01
		• Change R-UIM to CSIM	Editor	2.01
5.2.57	EF_{EPRL}	• Add encoding details	C.S0023-Dv2.0	2.01
		• Change “EPRL” to subscript	Editor	2.15
5.2.64	EF_{MMSN}	• Clarify encoding	C.S0023-Dv2.0	2.01
5.2.66	EF_{MMSICP}	• Change 1st to First and 2nd to Second	C.S0023-Dv2.0 Editor	2.01
		• Replace word “Second”	Editor	2.15
5.2.69	$EF_{AuthCapability}$	• Clarify that byte 3 is part of Authentication Capability field.	20101021-008	2.01
		• Add direct references to RFCs to list of references and replace by [69] through [74]	C.S0023-Dv2.0 Editor	2.01
5.2.70	EF_{3GCIK}	• Note that this is just the CK and IK produced by ‘3G Access AKA’ (and not by EAP AKA). • Revert “PIN” back to “ADM” for “UPDATE”.	20121015-007	2.05
5.2.76	EF_{GROUP_TAG}	• Correct reference to section of [7].	C.S0023-Dv2.0	2.01
5.2.77	$EF_{SPECIFIC\_TAG}$	• Correct reference to section of [7].	C.S0023-Dv2.0	2.01
5.2.78	EF_{CALL_PROMPT}	• Correct reference to section of [7].	C.S0023-Dv2.0	2.01
5.2.80	EF_{EST}	• Remove • characters	Editor	2.03
		• Remove row header “Contents” • Add EF names to column headers • Add “Service” to last column • Add borders to table	Editor	2.04
		• Simplification of the meaning of this file versus EF_{CSIM_ST}	20121210-025r1	2.06
5.2.84	EF_{SDN}	• Mark byte X+13 (EF_{CCP2} Record Identifier) as reserved (‘FF’). • Remove mention of where capability/configuration parameters are stored.	20130514-006	2.10
5.2.87	EF_{ICI}	• Mark byte X+13 (EF_{CCP2} Record Identifier) as reserved (‘FF’).	20130514-006	2.10
5.2.88	EF_{ICI}	• Mark byte X+13 (EF_{CCP2} Record Identifier) as reserved (‘FF’).	20130514-006	2.10
5.2.90	EF_{CCP2}	• Note that this EF contains no valid information. • For backwards compatibility recommend user of filler.	20130311-012	2.08
		• Remove text noting that this is still referenced	Editor	2.10
5.2.92	$EF_{AppLabels}$	• Emphasize that use of the labels by the ME is optional.	C.S0023-Dv2.0	2.01
		• Clarify encoding with 7 bit characters	20130711-005	2.12

Section	Name	Changes	Source	v
5.2.94	EF_{RC}	• Add borders to a table	Editor	2.01
		• Widen table borders so they don't disappear in PDF.	Editor	2.15
5.2.96	$EF_{MIPFlags}$	• Change "MobileIP" in title to "Mobile IP"	C.S0023-Dv2.0	2.01
		• Add reference to [23] (X.S0011) for Mobile IP 2002bis.	C.S0023-Dv2.0	2.01
		• Add borders to MIP_FLAGS table	Editor	2.01
		• Clarify meaning of both values of bit #1 of MIP_FLAGS	20130821-006	2.13
		• Use numeric reference number for RFC 2002	20130909-015r1	2.14
5.2.97	$EF_{3GPDUPPEXT}$	• Add new application type '8' for Administration (BIP). • Change reserved values to start at '9'.	C.S0023-Dv2.0	2.01
		• Put borders on encoding tables	Editor	2.01
5.2.99	$EF_{IPV6CAP}$	• Add borders to IPv6 Flags table.	Editor	2.01
5.2.102	$EF_{WAPBrowserCP}$	• Widen table borders so they don't disappear	Editor	2.15
5.2.103	$EF_{WAPBrowserBM}$	• Widen table borders so they don't disappear	Editor	2.15
5.2.104	$EF_{MMSConfig}$ (MMS Configuration)	• Refer to other EF directly by reference to EF name (MMSICP)	C.S0023-Dv2.0	2.01
5.2.105	EF_{JDL}	• Clarify encoding to ensure that EF can be longer than current URL.	20101007-021	2.01
		• Incorporate formatting and grammatical changes from C.S0023.	C.S0023-Dv2.0 Editor	2.01
5.2.106	EF_{ARR}	• Add this new EF	20130603-015	2.11
~~5.2.106~~5.3	DFs at the CSIM ADF	• Remove duplicate note.	20130514-006	2.10
5.4.3	EF_{ECCP}	• Change title to include all three EFs that are mentioned. Leave this section in for backwards compatibility. • Clarify that $DF_{PHONEBOOK}$ being referred to is from DF_{CDMA} not $DF_{TELECOM}$	20130514-006	2.10
		• Remove note saying that EF is referenced elsewhere	Editor	2.10
5.5.3	Contents of Files at the $DF_{MULTIMEDIA}$ level	• Change "MML" to subscript	Editor	2.15
6	Interworking			
7	Application Protocol			
7.1.1.1	CSIM Application Selection	• Change "R-UIM_ID" to "RUIMID"	20130409-005	2.09
7.1.2	Session Termination	• Remove information about AKA (3GCIK) update which does not happen	20121015-007	2.05
7.1.4	Emergency call codes request	• Describe the purpose of the ECC list. Remove mention of ME-specific emergency call codes.	20130311-010R3	2.08
7.1.11	RUIMID Request	• Change title from "R-UIM ID" to "RUIMID"	20130409-005	2.09

Section	Name	Changes	Source	v
7.1.12	ESN_MEID_ME update	• Capitalize name of command	20121015-008	2.05
		• Change “R-UIM” to UICC	Editor	2.07
7.2.1	AKA(3GCIK)	• Delete “Update” command which implies that the ME stores its copy of the 3G CK/IK in the card.	20121015-007	2.05
7.3.2	Dialing Numbers	• Simplify description of FDN handling now that service n32 is deprecated.	20121220-025r1	2.06
7.3.4	Capability Configuration Parameters	• Mark void and delete all text.	20130514-006	2.10
7.4	CCAT Procedures	• Add new call control services	20130311-035	2.08
8	Structure of Commands and Responses			
8.1.1	Coding of Class Byte	• Note that class bytes are defined in Table 1.	20121015-008	2.05
8.1.2	Coding of Instruction byte	• Add P1, P2 columns with values, where appropriate. • Include all sub-commands defined in C.S0065	20121015-008	2.05
		• Change “&” to “AND” in “S-SAFE VERIFICATION AND DECRYPTION” command.	Editor	2.05
8.1.3	Coding of Parameter bytes	• Indicate that these values are defined in Table 1 or Section 9.	20121015-008	2.05
8.1.4	Coding of Lc bytes	• Indicate that these values are defined in Table 1 or Section 9.	20121015-008	2.05
8.1.5	Coding of Data part	• Also partially defined in Section 9.	20121015-008	2.05
8.1.6	Coding of Le bytes	• Define field. • Also partially defined in Section 9.	20121015-008	2.05
8.2	Response APDU structure	• Also partially defined in Section 9.	20121015-008	2.05
9	Commands	• Various grammar and punctuation changes.	20121015-008	2.05
9.4.1	Security-related commands	• Indicate the SW codes for sequence errors • Add encoding for CONFIRM SSD command	20120718-007	2.04
		• Reverse Comprion contribution as SW codes should not be changed for the out-of-sequence error.	20121015-016	2.05
		• Capitalize command sub-command names (except references to R-UIM sub-commands). • Clearly define which parts of commands are defined in Table 1 and which parts in C.S0023 (R-UIM, reference [46]). • Delete unnecessary tables defining CLA, INS, P1, P2 etc.	20121015-008	2.05
		• Reverse Comprion change of CONFIRM SSD CLA from ‘00’ back to ‘8X’.	Editor	2.05
		• Standardize names of AKA commands	20121015-007	2.05
9.4.2	OTASP/OTAPA-related commands	• Capitalize command sub-command names (except references to R-UIM sub-commands). • Clearly define which parts of commands are defined in Table 1 and which parts in C.S0023 (R-UIM, reference	20121015-008	2.05

Section	Name	Changes	Source	v
		[46]). • Delete unnecessary tables defining CLA, INS, P1, P2 etc. • Replace incorrect references to Le (maximum length of message) with N4 (actual length of message).		
9.4.2.6	OTAPA Request	• Show unused portion of ESN field as RFU (as in R-UIM) • Show all '0' bits of octet 1 and 2 as 7 bit RFU fields. • Note that Start/Stop must be '1' in order to return RAND_OTAPA.	20121015-008	2.05
9.4.3	ESN Management Commands	• Capitalize command sub-command names (except references to R-UIM sub-commands). • Clearly define which parts of commands are defined in Table 1 and which parts in C.S0023 (R-UIM, reference [46]). • Correct STORE ESN_MEID_ME table. • Identify RFU bits for ESN_ME • Replace text description of fields by encoding tables.	20121015-008	2.05
9.4.4	Packet-Data security-related Commands	• Capitalize command sub-command names (except references to R-UIM sub-commands). • Clearly define which parts of commands are defined in Table 1 and which parts in C.S0023 (R-UIM, reference [46]).	20121015-008	2.05
9.4.5	BCMCS-related Commands	• Clearly define which parts of commands are defined in Table 1 and which parts in C.S0023 (R-UIM, reference [46]).	20121015-008	2.05
9.4.6	Application Authentication Commands	• Capitalize command sub-command names (except references to R-UIM sub-commands). • Clearly define which parts of commands are defined in Table 1 and which parts in C.S0023 (R-UIM, reference [46]).	20121015-008	2.05
9.4.7	AKA-related Commands	• Capitalize command sub-command names (except references to R-UIM sub-commands). • Clearly define which parts of commands are defined in Table 1 and which parts in C.S0023 (R-UIM, reference [46]).	20121015-008	2.05
		• Reference section 9.4.1.4 for the 3G ACCESS AKA and EAP AKA commands.	20121015-007	2.05
9.4.8	LCS-related Commands	• Capitalize command sub-command names (except references to R-UIM sub-commands). • Clearly define which parts of commands are defined in Table 1 and which parts in C.S0023 (R-UIM, reference [46]).	20121015-008	2.05
		• Change "&" to "AND" in "S-SAFE VERIFICATION AND DECRYPTION" command.	Editor	2.05
10	Description of Services-Related Procedure			
10.1.1.1	CSIM	• Change command names to all upper case • Correct grammar	20121015-008	2.05
		• Change "&" to "AND" in "S-SAFE VERIFICATION AND DECRYPTION" command.	Editor	2.05
10.1.2	Key Management	• Capitalize Command Name	20121015-008	2.05

Section	Name	Changes	Source	v
Annex C	CSIM Application Session Activation/Termination	• Put title on title line • Change “initialization” in Figure 1 to “initialization” (2 times, not change marked).	20130409-005	2.09
Annex E	Summary of CSIM Files	• Change $EF_{3GPDOPM}$ Update access from ADM to PIN	20120412-006	2.02
		• This change of Update access has been reversed.	20121015-007	2.05
		• Change “NEVER” to “Never” in several places	Editor	2.09
		• Correct EF_{HIDDEN_KEY} to $EF_{HiddenKey}$	20130409-005	2.09
Table 3	Annex E	• Reserve file id assigned to EF_{CCP2}	20130514-006	2.10

Contents

Figures

Tables

FOREWORD

(This foreword is not part of this specification).

The present document defines the cdma2000®[1] (CSIM) application. This application resides on the UICC, an IC card specified in [45]. In particular, [45] specifies the application independent properties of the UICC/terminal interface such as the physical characteristics and the logical structure. This document also inherits many of the Elementary File types and other characteristics from the R-UIM specification [46].

[1] cdma2000® is the trademark for the technical nomenclature for certain specifications and standards of the Organizational Partners (OPs) of 3GPP2. Geographically (and as of the date of publication), cdma2000® is a registered trademark of the Telecommunications Industry Association (TIA-USA) in the United States.

1. INTRODUCTION

~~The present document~~This specification defines the cdma2000 (CSIM) application. This application resides on the UICC, an IC card specified in [45]. In particular, [45] specifies the application independent properties of the UICC/terminal interface such as the physical characteristics and the logical structure.

1.1 Scope

The present document defines the cdma2000 application for cdma2000 network operation.

The present document specifies:

- Specific command parameters;
- File structures;
- Security functions;
- Interworking with other Applications (ISIM, USIM, etc.~~....~~) on UICC
- Application protocol to be used on the interface between UICC (cdma2000 application) and ME.

This is to ensure interoperability between a CSIM and an ME independently of the respective manufacturer, card issuer or operator.

The present document does not define any aspects related to the administrative management phase of the cdma2000 application. Any internal technical realization of either the cdma2000 application or the ME is only specified where these are reflected over the interface. The present document does not specify any of the security algorithms that may be used.

1.2 Requirements Language

"Shall" and "shall not" identify requirements to be followed strictly to conform to this document and from which no deviation is permitted. "Should" and "should not" indicate that one of several possibilities is recommended as particularly suitable, without mentioning or excluding others, that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is discouraged but not prohibited. "May" and "need not" indicate a course of action permissible within the limits of the document. "Can" and "cannot" are used for statements of possibility and capability, whether material, physical or causal.

1.3 References

The following standards are referenced in this text. At the time of publication, the editions indicated were valid. All standards are subject to revision, and parties to agreements based upon this document are encouraged to investigate the possibility of applying the most recent editions of the standards indicated below. ANSI and TIA maintain registers of currently valid national standards published by them.

1.3.1 Normative: References

1. 3GPP2 C.S0001-~~D v2.0~~E ~~v1.0~~v3.0, *Introduction to cdma2000 Spread Spectrum Systems*, ~~October 2005~~September 2009June 2011.
2. Reserved.~~3GPP2 C.S0002-D v2.0E v1.0, *Physical Layer Standard for cdma2000 Spread Spectrum Systems*, September 2009October, 2005.~~
3. Reserved.
4. Reserved.~~3GPP2 C.S0004-E v1.0D v2.0, *Signaling Link Access Control (LAC) Standard for cdma2000 Spread Spectrum Systems*, September, 2009October 2005.~~
5. 3GPP2 C.S0005-~~E v3.0D v2.0~~, *Upper Layer (Layer 3) Signaling Standard for cdma2000 Spread Spectrum Systems*, June 2011~~October 2005~~.
6. Reserved.
7. 3GPP2 C.S0016-~~C v2.0~~D v2.0, *Over-the-Air Service Provisioning of Mobile Stations in Spread Spectrum Systems*, April 2012. ~~October 2008.~~
8. C.S0015-B v2.0, *Short Message Service for Spread Spectrum Systems*, October 2005.
9. ITU-T Recommendation E.212, "Identification Plan for Land Mobile Stations", ~~November 1998~~May 2008.
10. Reserved.
11. Reserved.
12. Reserved
13. Reserved
14. TIA-95-B, *Mobile Station - Base Station Compatibility Standard for Wideband Spread Cellular Systems*, October 2004.
15. 3GPP2 X.S0004-E ~~V2~~v9.0.~~0~~, *Mobile Application Part*, ~~July, 2005~~June 2009.
16. Reserved.~~TIA/EIA/IS-91-A, *Base Station – Mobile Station Compatibility Specification for 800 MHz Cellular, Auxiliary, and Residential Services*, November 1999.~~

17. 3GPP TS 51.011 V4.15.0, *Specification of the Subscriber Identity Module-Mobile Equipment (SIM-ME) Interface*, June 2005.

18. ETSI TS 102 221 ~~V8.1.0~~V10.0.0, *Smart cards; UICC-Terminal Interface; Physical and logical Characteristics*, ~~April 2009~~December 2011.

19. Reserved.

20. Reserved.~~3GPP2 S.S0053-0 v2.0 *Common Cryptographic Algorithms*, May 2009.~~

21. Reserved.

22. Reserved.

23. 3GPP2 X.S0011-~~D v2.0~~E v1.0, cdma2000 *Wireless IP Network Standard*, ~~November 2008~~November 2009.

24. Reserved.~~IETF RFC 3344, *IP Mobility Support*, August 2002.~~

25. Reserved.~~IETF RFC 2794, *Mobile IP Network Access Identifier Extension for IPv4*, March 2000.~~

26. Reserved.~~IETF RFC 2865, *Remote Authentication Dial In User Service (RADIUS)*, June 2000.~~

27. Reserved.~~IETF RFC 4721, *Mobile IPv4 Challenge/Response Extensions*, January 2007.~~

28. 3GPP2 C.S0024-~~B~~ C ~~v2.0~~v2.0, *cdma2000 High Rate Packet Data Air Interface Specification*, ~~April 2007~~July 2011.

29. Reserved.~~3GPP2 A.S0008-C v2.0, *Interoperability Specification (IOS) for High Rate Packet Data (HRPD) Access Network Interfaces*, January 2009.~~

30. 3GPP TS 31.102 ~~V8.6.0~~V11.0.0, *Characteristics of the Universal Subscriber Identity Module (USIM) application*, ~~June 2009~~October 2010.

31. Reserved.~~3GPP TS 31.103 V8.1.0, *Characteristics of the IP Multimedia Services Identity Module (ISIM) Application*, June 2009.~~

32. Reserved.~~3GPP2 X.S0013-000-B v1.0, *All-IP Core Network Multimedia Domain - Overview* , December 2007.~~

33. Reserved.~~IETF RFC 3261, *SIP: Session Initialization Protocol*, June 2002.~~

34. Reserved.~~IETF RFC 4282, *The Network Access Identifier*, December 2005.~~

35. Reserved.

36. 3GPP2 S.S0083-A v1.0, *Broadcast-Multicast Service Security Framework*, September 2004.

37. 3GPP2 X.S0016-200-A v1.0, *MMS Stage-2, Functional Description*, February 2006.

38. 3GPP TS 23.038 ~~V8.2.0~~V9.1.1, *Alphabets and language-specific information*, ~~September 2008~~February 2010.

39. 3GPP2 X.S0016-310 v2.0, *MMS MM1 Stage-3 Using OMA/ WAP*, July 2004.

40. 3GPP2 X.S0016-311-0 v1.0, *MMS MM1 Stage-3 Using M-IMAP for message submission and retrieval, May 2003.*

41. 3GPP2 X.S0016-312-0 v1.0, *MMS MM1 Stage-3 Using SIP*, July 2004.

42. Reserved.~~3GPP2 S.S0055-A V4.0, *Enhanced Cryptographic Algorithms*, January 2008.~~

43. Reserved.

44. 3GPP2 C.S0068-0 v1.0, *ME Personalization*, June 2006.

45. 3GPP2 C.S0074-A v1.0, *UICC-Terminal Interface Physical and Logical Characteristics for cdma2000 Spread Spectrum Systems,* ~~tbd 2009~~January 2010.

46. 3GPP2 C.S0023-~~C v2.0~~D v2.0, *Removable User Identity Module for Spread Spectrum Systems*, ~~October 2008~~December 2011.

47. 3GPP2 C.S0035-A v2.0, *CDMA Card Application Toolkit (CCAT)*, August 2007.

48. ETSI TS 101 220 ~~V8.4.0~~V11.0.0, *Smart cards; ETSI numbering system for telecommunication application providers*, ~~April 2009~~June 2011.

49. Reserved.~~3GPP TS 11.11 V8.14.0, "Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) Interface", June 2007.~~

50. S.S0110-0 v1.0, *IP-based Location Services Security Framework*, March 2006.

51. Reserved.

52. Reserved.~~IETF RFC 5246, The TLS Protocol Version 1.2, August 2008.~~

53. ISO/IEC 7816-4, *Identification cards - Integrated circuit~~(s)~~ cards ~~with contacts~~, Part 4: ~~Interindustry~~ Organization, security and commands for interchange*, ~~October 2008~~January 2005.

54. ETSI TS 102 222 V7.1.0, *Administrative commands for telecommunications applications*, February 2007.

55. Reserved.

56. Reserved.

57. Reserved.

58. ITU E.118 The international telecommunication charge card, ~~February 2001~~May, 2006.

59. ITU X.509, *Public-key and attribute certificate frameworks*, ~~August 2005~~November 2008.

60. ITU X.690, *ASN.1 encoding rules*: *Specification of Basic Encoding Rules (BER), Canonical Encoding Rules (CER) and Distinguished Encoding Rules (DER)*, November 2008.

~~*Editor's Note: The above document is a work in progress and should not be referenced unless*~~

~~*and until it is approved and published. Until such time as this Editor's Note is removed, the inclusion of the above document is for informational purposes only.*~~

61. IETF RFC 2315, *PKCS #7: Cryptographic Message Syntax Version 1.5*, March 1998.

62. RSA PKCS #12 v1.0, *Personal Information Exchange Syntax*, March 1998.

63. IETF RFC 4489, *A Method for Generating Link-Scoped IPv6 Multicast Addresses*, April 2006.

64. IETF RFC 1738, *Uniform Resource Locators (URL)*, December 1994.

65. 3GPP2 C.S0017-012-A ~~v1~~v2.0, *Data Service Options for Spread Spectrum Systems: Service Options 33 and 66*, ~~July, 2004~~June 2006.

66. ISO/IEC 7816-5, *Identification cards -- Integrated circuit cards -- Part 5: Registration of application providers*, December 2004.

67. IETF RFC 3629, *UTF-8, a transformation format of ISO 10646*. November 2003.

68. 3GPP2 C.S0057-E v1.0, *Band Class Specification for cdma2000 Spread Spectrum Systems*, October 2010.

69. IETF RFC 2195. *IMAP/POP AUTHorize Extension for Simple Challenge/Response*. September 1997.

70. IETF RFC 2617. *HTTP Authentication: Basic and Digest Access Authentication*. June 1999.

71. IETF RFC 3310. *Hypertext Transfer Protocol (HTTP) Digest Authentication Using Authentication and Key Agreement (AKA)*. September 2002.

72. IETF RFC 2831. *Using Digest Authentication as a SASL Mechanism*. May 2000.

73. IETF RFC 2444. *The One-Time-Password SASL Mechanism*. October 1998.

74. IETF RFC 2222. *Simple Authentication and Security Layer (SASL)*. October 1997.

75. IETF RFC 2002. *IP Mobility Support*. October 1996.

1.3.2 Informative~~:~~ References

1. 3GPP2 C.R1001-~~F~~ G v1.0, *Administration of Parameter Value Assignments for cdma2000 Wideband Spread Spectrum Standards*, ~~December 2006~~June 2009.

2. 3GPP TS 31.101~~:~~, ~~"~~*UICC-Terminal Interface, Physical and Logical Characteristics*~~"~~.

3. 3GPP TS 23.140~~:"~~, *Multimedia Messaging Service (MMS); Functional description; stage 2*~~"~~.

2. VOID

3. VOID

4. DEFINITIONS, SYMBOLS, ABBREVIATIONS AND CODING CONVENTIONS

For the purposes of the present document, the following terms and definitions apply:

AID. ISO/IEC 7816 Application Identifier. See [48], [53] and [66]~~[53/66]~~.

Card Session. See [17].

CDMA Session. That part of the *Card Session* dedicated to the CDMA operation.

CSIM. cdma2000 Subscriber ~~Identify~~ Identity Module. cdma2000 Application residing on the UICC, an IC card specified in [45].

ECC (Emergency Call Code). A number, that when dialed by the user, is to be treated as an emergency call.

ESN (Electronic Serial Number). A 32-bit number that may be the ESN_ME or UIM_ID.

ESN_ME. A 32-bit number that may be a unique value assigned to a mobile station or a non-unique value derived from the MEID_ME (~~pseudo-~~pESN).

EUIMID. Expanded UIMID. SF_EUIMID or LF_EUIMID.

ICCID. The International Charge Card Identifier. See [58].

LCS. Location services.

LCS Root Key. LCS related parameter. See [50].

LF_EUIMID. Long form EUIMID, the ICCID.

MEID (Mobile Equipment Identifier). A 56-bit number (14 hexadecimal digits) that may be the MEID_ME or SF_EUIMID.

MEID_ME. A 56-bit number uniquely assigned to a mobile station by a manufacturer.

MMSS. Multi-Mode System Selection. See [7].

PIX. Proprietary application Identifier eXtension. See [48], [53] and [66]~~[53/66]~~.

pESN. A 32-bit number derived from MEID and used in place of ESN. See section 2.3.2.2.1 of [5].

pUIMID. A 32-bit number derived from EUIMID and used in place of UIMID. See section 5.2.17 (EFRUIMID).

RID. Registered Application Provider Identifier. See [48], [53] and [66]~~[53/66]~~.

R-UIM. Removable User Identity Module residing on a Non-UICC based platform, as specified in [46].

SF_EUIMID. A 56-bit number uniquely assigned to an R-UIM using the same format as MEID_ME and assigned from the same numbering space.

S-SAFE. Secure Store-And-Forward-Encapsulation. LCS related parameter. See [50].

TLS. Transport Layer Security.

UI. User Interface.

UIM_ID. A 32-bit electronic identification number unique to an R-UIM or a non-unique value derived from the EUIMID (~~pseudo-~~pUIM_ID).

All other definitions, symbols, abbreviations applicable to the R-UIM specified in [46] and UICC specified in [45] are applicable here.

~~The AID of CSIM is defined in [48/53] and is stored in EF_{DIR}. It is composed of the RID code 0xA000000343 and the PIX code, of which the first four digits are the 3G App Code 0x1002 indicating the "3GPP2 CSIM" application.~~

4.1 Coding Conventions

All unused, allocated memory shall be set to zero unless otherwise specified. RFU bits shall be set to zero and may be used in the future for additional parameters. Reserved bits shall be set to zero unless otherwise specified and shall not be used in the future for additional parameters. The ME shall ignore the state of all RFU and Reserved bits.

Single quotes indicate binary or hexadecimal values (e.g. '00000001' or 'A0'). Valid elements for hexadecimal values are the digits '0' to '9' and 'A' to 'F' (representing the values 10 through 15).

4.1.1 CSIM Status Codes

Status codes sent by the CSIM to an ME via octets SW1 and SW2 are defined in [18] except for SW1='98', SW2='34' (originally defined in [17]) which means, "Error, out of sequence".

4.2 Parameters Stored Temporarily in the CSIM

The following parameters with subscript "s" indicate a value stored temporarily in the CSIM:

NAM_LOCK_S – A network controlled status of the SPASM protection of the active NAM for the subsequent OTAPA session – temporarily stored in the CSIM.

SPC_S – Service Programming Code temporarily stored in the CSIM if the Service Programming Lock feature is supported by the CSIM.

SSD_S – A secret 128-bit pattern for the Shared Secret Data temporarily stored in the CSIM.

5. FILES

This section specifies the EFs for the CDMA operation defining access conditions, contents and coding.

A file is associated with attributes that depending of the file type indicates how data is to be accessed e.g. file size, record length etc. Although in the present document some files and data items stored in a file are indicated as having a fixed length; when reading such structures the ME shall derive the length of the data item from the attributes provided in the file information i.e. not use the fixed value specified for the file in the present document. Although the ME is able to read the entire structure it should only use those elements in the data item which is recognized by the ME.

For any EF, if the SFI (Short (elementary) Form Indicator) is not indicated in the description of the file, then it is not allowed to assign an SFI. If in the description of the file an SFI value is indicated, then the file shall support SFI. The SFI value shall be assigned by the card issuer. It is mandatory for EFs stating an SFI value ('YY') in the description of their structure to provide an SFI. For files where in the file description the SFI is indicated as 'Optional', then the file may support an SFI.

References [1] and [14] store parameters in several different types of memory.

- Variables stored in permanent memory use the subscript "p".
- Variables stored in semi-permanent memory use the subscript "s-p".
- Variables temporarily stored (including those parameters defined in Sec. 3.2 which use the subscript s).

5.1 Contents of files at the MF level

There are four application independent EFs at the Master File (MF) level as specified in [45], i.e.: EF_{ICCID}, EF_{DIR}, EF_{PL} and EF_{ARR}.

EF_{DIR} stores the AID of CSIM, defined in section 4 of [48]. The AID is composed of the RID code 0xA000000343 ('3GPP2') and the PIX code, of which the first four digits are the 3G App Code 0x1002 indicating the "3GPP2 CSIM" application, defined in Annex M of [48]. The remaining digits should be formatted according to Annex F of [48].

Multiple CSIM applications can be distinguished by using different values in digits 21 and 22 of AID for each application.

~~See section 5.2.91 for some additional restrictions on the contents of EF_{ICCID}.~~

5.1.1 EF_{ICCID} (ICC Identification)

EF_{ICCID} is as defined in [18] with the following restrictions:

- This EF shall contain 18 digits of the actual ICCID followed by the check digit and a single 0xF filler digit.
- The ICCID shall be globally unique, using an Issuer Identifier Number registered with the ITU-T as specified in [58].
- If the long form of the EUIMID is chosen, the ICCID is the LF_EUIMID.

5.2 Contents of files at the CSIM ADF (Application DF) level

5.2.1 EF_{COUNT} (Call Count)

This EF stores the value of Call Count, COUNTs-p.

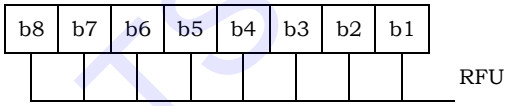
Identifier: '6F21'	Structure: cyclic	Mandatory
Record Length: 2 bytes	Update activity: high	
Access Conditions: READ PIN UPDATE PIN INCREASE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 – 2	COUNTs-p	M	2 bytes

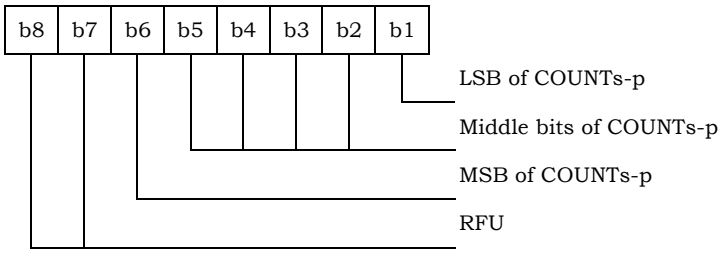
COUNTs-p is contained in the least significant 6 bits of the two-byte field.

Coding:

Byte 1:



Byte 2:



5.2.2 EF_{IMSI_M} (IMSI_M)

This EF stores the five components of IMSI_M.

Identifier: '6F22'	Structure: transparent	Mandatory
SFI: '04'		
File size: 10 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE PIN		

Bytes	Description	M/O	Length
1	$IMSI\_M\_CLASS_p$	M	1 byte
2 – 3	IMSI_M_S2 from $IMSI\_M\_S_p$	M	2 bytes
4 – 6	IMSI_M_S1 from $IMSI\_M\_S_p$	M	3 bytes
7	$IMSI\_M\_11\_12_p$	M	1 byte
8	IMSI_M_PROGRAMMED/ $IMSI\_M\_ADDR\_NUM_p$	M	1 byte
9 –10	MCC_M_p	M	2 bytes

IMSI_M_CLASSp - Class assignment of the IMSI_M.

IMSI_M_ADDR_NUMp - Number of IMSI_M address digits.

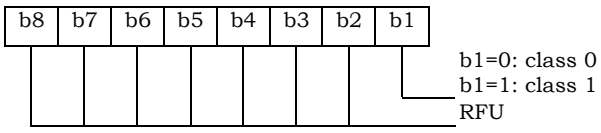
MCC_Mp - Mobile country code.

IMSI_M_11_12p - 11th and 12th digits of the IMSI_M.

IMSI_M_Sp - The least significant 10 digits of the IMSI_M.

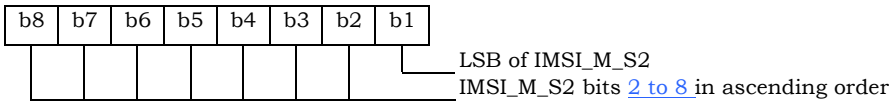
Coding:

Byte 1:

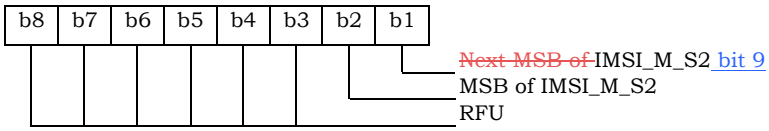


Byte 2, byte 3, byte 4, byte 5 and byte 6 are encoded as described in Section 2.3.1.1 of [5] and ~~[14],~~ Section 6.3.1.1 of [14], "Encoding of IMSI_M_S and IMSI_T_S". IMSI_M_S2 contains the most significant digits of IMSI_M_S and IMSI_M_S1 contains the least significant digits of IMSI_M_S as described in Figure 2.3.1.-2 of [5] and Figure 6.3.1-2 of [14].

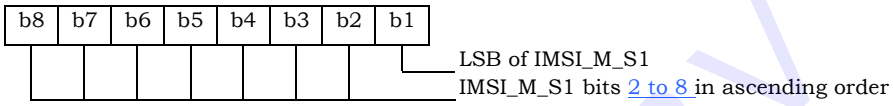
Byte 2:



Byte 3:

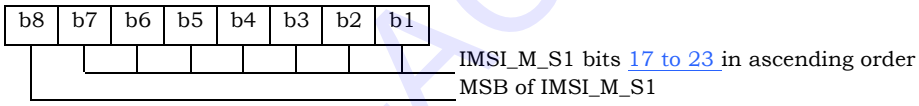


Byte 4:



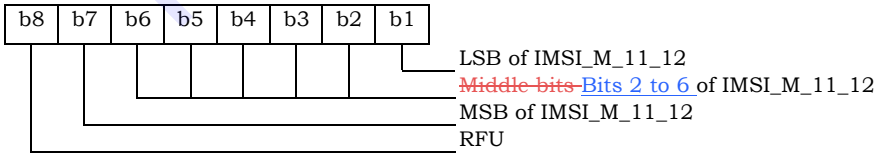
Byte 5:

Byte 6:



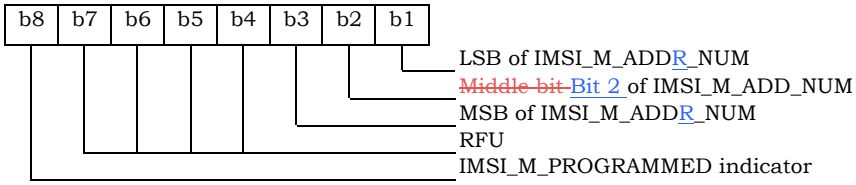
Byte 7 is encoded as described in ~~[14],~~ Section 2.3.1.2 of [5] and Section 6.3.1.2 of [14], "Encoding of IMSI_M_11_12 and IMSI_T_11_12".

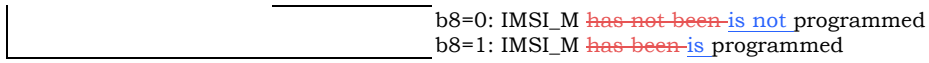
Byte 7:



Byte 8 is the binary equivalent of the IMSI_M_ADDR_NUM, as described in ~~[14],~~Section 2.3.1 of [5] and Section 6.3.1 or [14], "Mobile Station Identification Number".

Byte 8:

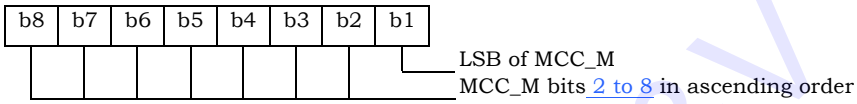




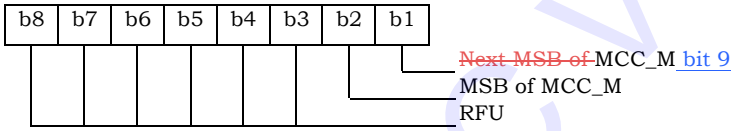
IMSI_M_PROGRAMMED shall be set to '1' if an IMSI_M has beenis programmed (IMSI_M would contain a MIN for systems that comply with [14]); if an IMSI_M has not been programmed, otherwise it shall be set to '0'. If OTASP is used to update this EF, see section 8.4.2.2 COMMIT. See [5] or [14] for details on IMSI_M programming.

Byte 9 and byte 10 are encoded as described in [14]Section 2.3.1.3 of [5] and Section 6.3.1.3 of [14], "Encoding of the MCC_M and MCC_T".

Byte 9:



Byte 10:



For CSIM applications in systems that comply with [5] or [14], the parameter "MIN" is stored in EF_{IMSI_M}. For these instances, the 10 bits of "MIN2" are stored in bytes 2 and 3, with the coding shown above, while the 24 bits of "MIN1" are stored in bytes 4, 5, and 6.

The selection of IMSI_M or IMSI_T for use in the authentication process shall be in accordance with [14] Section 6.3.12.1 and [5] Section 2.3.12.1, which stipulate that the "MIN" portion of IMSI_M shall be used as an input parameter of the authentication calculation if IMSI_M is programmed and that a 32-bit subset of IMSI_T shall be used if only IMSI_T has been programmed.

5.2.3 EF_{IMSI_T} (IMSI_T)

This EF stores the five components of IMSI_T.

Identifier: '6F23'	Structure: transparent	Mandatory
SFI: '05'		
File size: 10 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE PIN		

Bytes	Description	M/O	Length
1	$IMSI\_T\_CLASS_p$	M	1 byte
2 – 3	IMSI_T_S2 from $IMSI\_T\_S_p$	M	2 bytes
4 – 6	IMSI_T_S1 from $IMSI\_T\_S_p$	M	3 bytes
7	$IMSI\_T\_11\_12_p$	M	1 byte
8	IMSI_T_PROGRAMMED/ $IMSI\_T\_ADDR\_NUM_p$	M	1 byte
9 –10	MCC_T_p	M	2 bytes

All byte descriptions, encodings and reference sections in [5] and [14] are identical to those described in Section 5.2.2 EF_{IMSI_M}, except that all references to "IMSI_M" shall apply to "IMSI_T". EF_{IMSI_T} is not used to store a MIN.

5.2.4 EF_{TMSI} (TMSI)

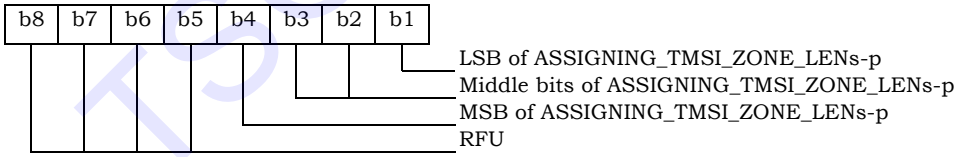
This EF stores the Temporary Mobile Station Identity (TMSI). TMSI is assigned by the serving network and consists of 4 components, i.e.: ASSIGNING_TMSI_ZONE_LENs-p, ASSIGNING_TMSI_ZONEs-p, TMSI_CODEs-p, and TMSI_EXP_TIMEs-p.

Identifier: '6F24'	Structure: transparent	Mandatory
SFI: '06'		
File size: 16 bytes	Update activity: high	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE PIN		

Bytes	Description	M/O	Length
1	$ASSIGNING\_TMSI\_ZONE\_LEN_{s-p}$	M	1 byte
2 – 9	$ASSIGNING\_TMSI\_ZONE_{s-p}$	M	8 bytes
10 – 13	$TMSI\_CODE_{s-p}$	M	4 bytes
14 – 16	$TMSI\_EXP\_TIME_{s-p}$	M	3 bytes

Coding:

Byte 1:



Bytes 2 through 9 store the (up to) 8-octet TMSI Zone as described in Section 2.3.15 of [5] and Section~~s~~ 6.3.15~~, 6.3.15.1 and 6.3.15.2~~ of [14]. These sections are entitled "Temporary Mobile Station Identity", "Overview" and "TMSI Assignment Memory" respectively. In each case the lowest-order octet shall be stored in the lowest-order byte (i.e., byte 2) of each set of contiguous 8 bytes, and successively higher octets stored in the next highest order bytes. Unused bytes shall be set to '00'.

Bytes 10 through 13 store the (2 to 4 octet) TMSI Code as described in the sections of [5] and [14] referenced above. In each case the lowest-order octet shall be stored in the lowest-order byte (i.e., byte 10) of each set of contiguous 4 bytes, and successively higher octets stored in the next highest order bytes. Unused bytes shall be set to '00'.

Bytes 14 through 16 store the TMSI Expiration Time as described in the sections of [5] and [14] referenced above. In each case the lowest-order octet shall be stored in the lowest-order byte (i.e.,

byte 14) of each set of contiguous 3 bytes, and successively higher octets stored in the next highest order bytes.

5.2.5 EF_{AH} (Analog Home SID)

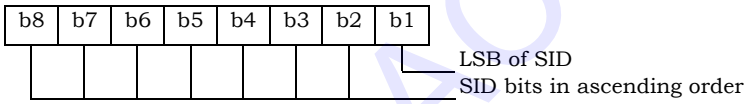
This EF identifies the home SID when the mobile station is operating in the analog mode.

Identifier: '6F25'	Structure: transparent	~~Mandatory~~Optional
File size: 2 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

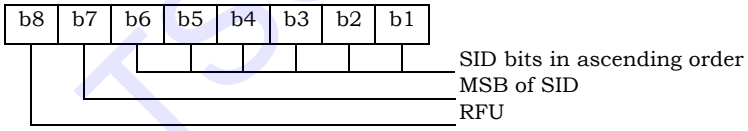
Bytes	Description	M/O	Length
1-2	Analog home SID ($HOME\_SID_p$)	M	2 bytes

Coding:

Byte 1:



Byte 2:



5.2.6 EF_{AOP} (Analog Operational Parameters)

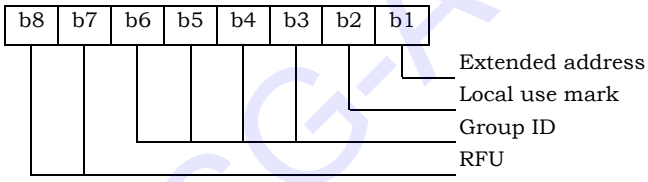
This EF includes the Extended Address bit (EXp), the Local Use Mark (LCM) and the Group ID (GID) field.

Identifier: '6F26'	Structure: transparent	~~Mandatory~~Optional
File size: 1 byte	Update activity: low	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Analog Operational Parameters (EX_p, LCM, GID)	M	1 byte

Coding:

Byte 1:



5.2.7 EF_{ALOC} (Analog Location and Registration Indicators)

This EF stores parameters related to Autonomous Registration memory (NXTREGs-p and SIDs-p) as well as the Location Area memory (LOCAIDs-p and PUREGs-p).

Identifier: '6F27'	Structure: transparent	~~Mandatory~~Optional
File size: 7 bytes	Update activity: high	

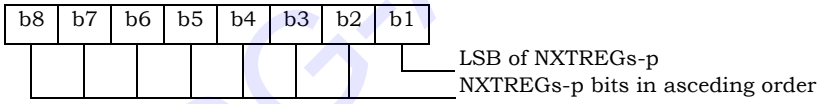
Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

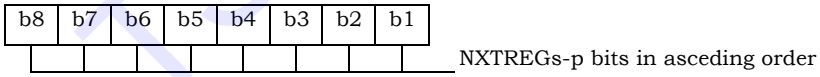
Bytes	Description	M/O	Length
1-3	$NXTREG_{s-p}$	M	3 bytes
4-5	SID_{s-p}	M	2 bytes
6-7	$LOCAID_{s-p}$, $PUREG_{s-p}$	M	2 bytes

Coding:

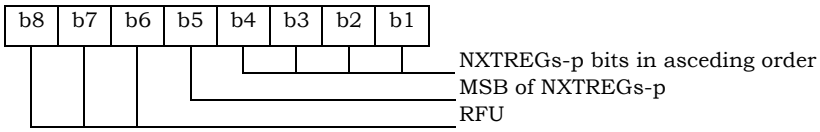
Byte 1:



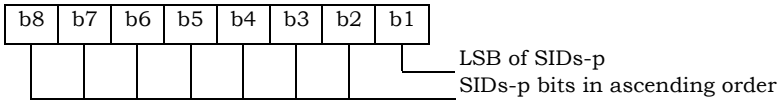
Byte 2:



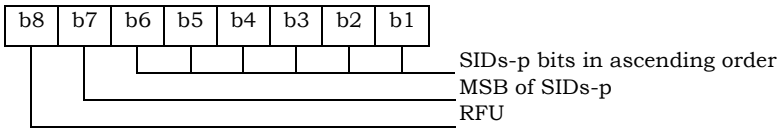
Byte 3:



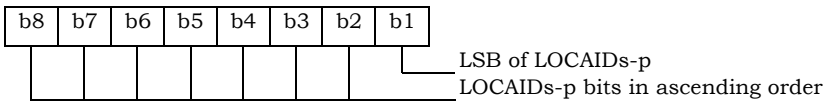
Byte 4:



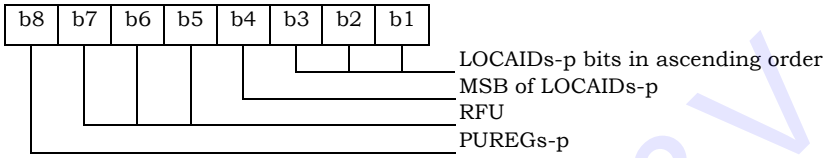
Byte 5:



Byte 6:



Byte 7:



5.2.8 $EF_{CDMAHOME}$ (CDMA Home SID, NID)

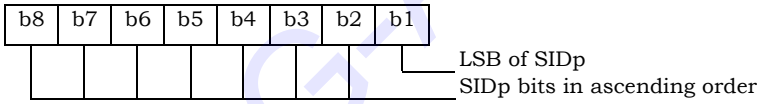
This EF identifies the home SID and NID when the mobile station is operating in the CDMA mode.

Identifier: '6F28'	Structure: linear fixed	Mandatory
SFI: '0C'		
Record length: 5 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

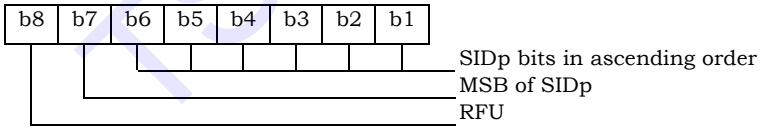
Bytes	Description	M/O	Length
1 – 2	CDMA Home SID (SID_p)	M	2 bytes
3 – 4	CDMA Home NID (NID_p)	M	2 bytes
5	Band Class	M	1 byte

Coding:

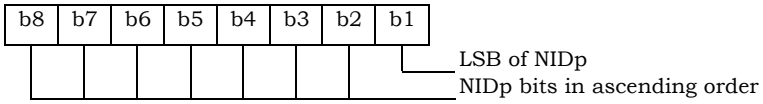
Byte 1:



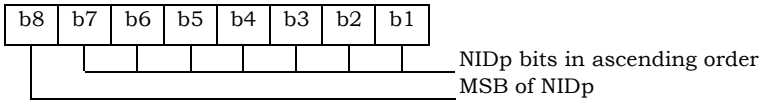
Byte 2:



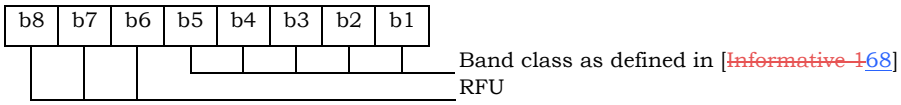
Byte 3:



Byte 4:



Byte 5:



5.2.9 EF_{ZNREGI} (CDMA Zone-Based Registration Indicators)

This EF stores the zone-based registration list "ZONE_LIST". The list includes a REG_ZONE and a corresponding SID, NID pair. Details are described in sections titled "Registration Memory", "Zone-Based Registration" and "Registration Procedures" of ~~[15/14]~~[5] and [14].

Identifier: '6F29'	Structure: linear fixed	Mandatory
Record length: 8 bytes	Update activity: high	

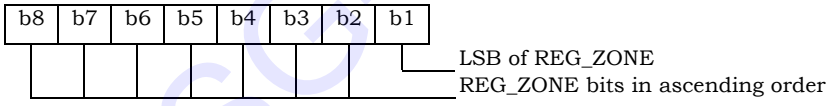
Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

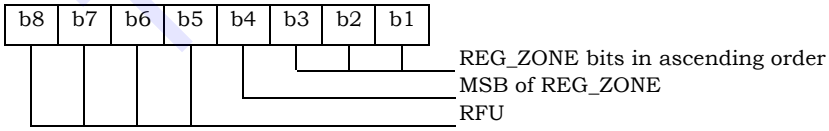
Bytes	Description	M/O	Length
1 – 2	REG_ZONE	M	2 bytes
3 – 4	SID	M	2 bytes
5 – 6	NID	M	2 bytes
7 – 8	RFU	M	2 bytes

Coding:

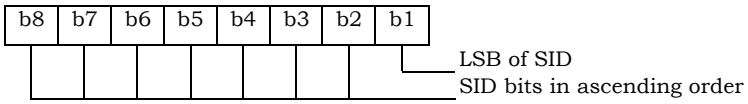
Byte 1:



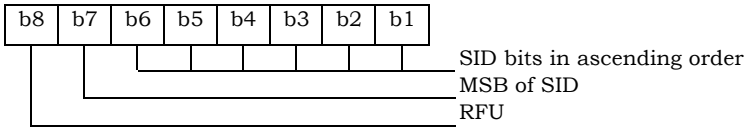
Byte 2:



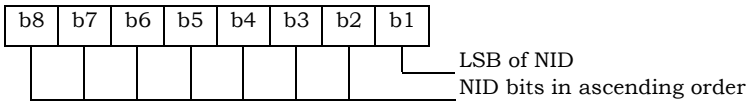
Byte 3:



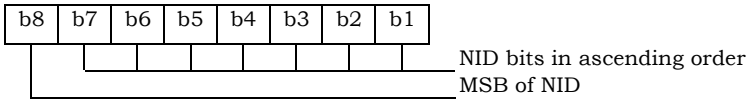
Byte 4:



Byte 5:



Byte 6:



5.2.10 EF_{SNREGI} (CDMA System-Network Registration Indicators)

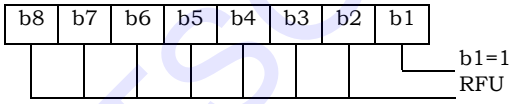
This EF stores the SID and NID of the wireless system in which the mobile station last registered. This is described in sections of [5] and [14] titled "Registration Memory" and "Zone-Based Registration", respectively.

Identifier: '6F2A'	Structure: transparent	Mandatory
SFI: '0D'		
File size: 7 bytes	Update activity: high	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

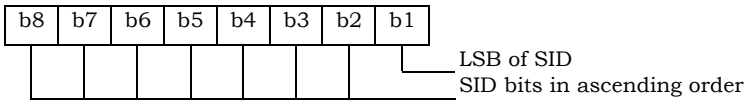
Bytes	Description	M/O	Length
1	N, size of SID/NID list (N=1)	M	1 byte
2 – 3	SID	M	2 bytes
4 – 5	NID	M	2 bytes
6 – 7	RFU	M	2 bytes

Coding:

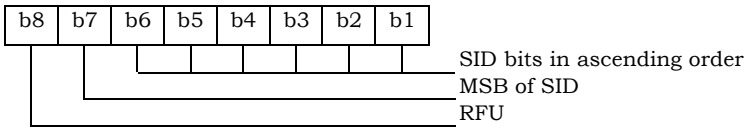
Byte 1:



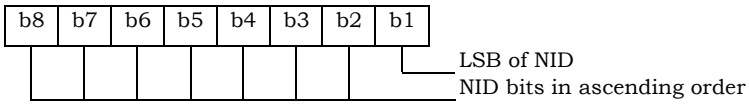
Byte 2:



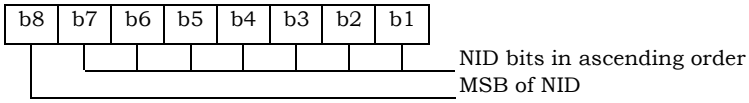
Byte 3:



Byte 4:



Byte 5:



5.2.11 $EF_{DISTREGI}$ (CDMA Distance-Based Registration Indicators)

This EF stores the Base Station Latitude (BASE_LAT_REG), the Base Station Longitude (BASE_LONG_REG) and the Registration Distance (REG_DIST_REG) of the base station to which the first access probe (for a Registration Message, Origination Message or Page Response Message) was transmitted after entering the System Access State.

Identifier: '6F2B'	Structure: transparent	Mandatory
File size: 8 bytes	Update activity: high	

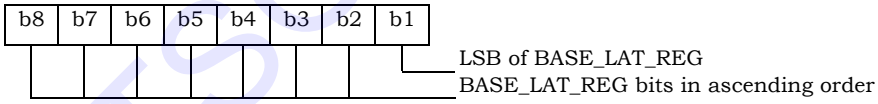
Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

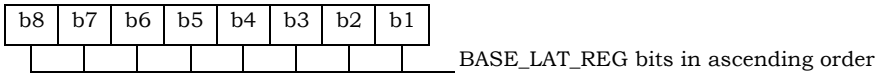
Bytes	Description	M/O	Length
1-3	BASE_LAT_REG	M	3 bytes
4-6	BASE_LONG_REG	M	3 bytes
7-8	REG_DIST_REG	M	2 bytes

Coding:

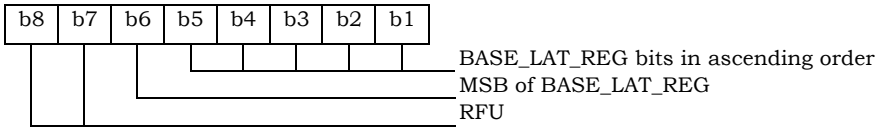
Byte 1:



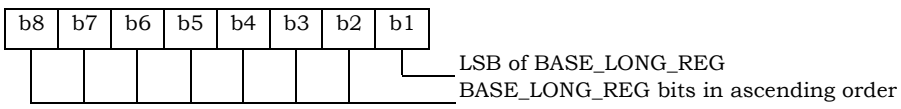
Byte 2:



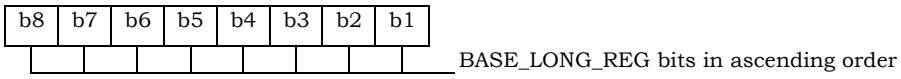
Byte 3:



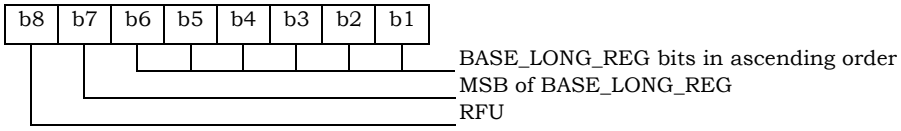
Byte 4:



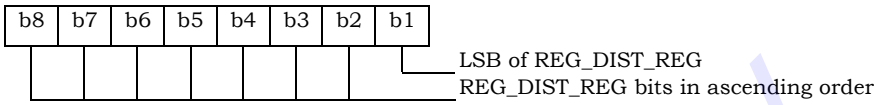
Byte 5:



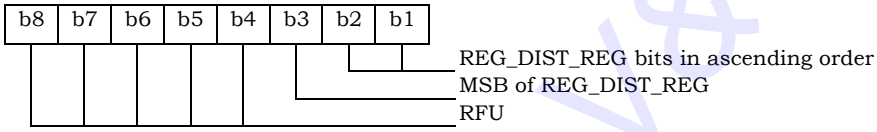
Byte 6:



Byte 7:



Byte 8:



NOTE: The parameters for Distance-Based Registration are described in ~~[14],~~ Section 2.6.5.1.4 of [5] and Section 6.6.5.1.4 of [14].

5.2.12 EF_{ACCOLC} (Access Overload Class ACCOLCp)

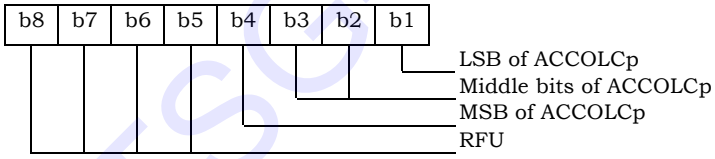
This EF defines the access overload class for the mobile station. This access overload class identifies which overload class controls access attempts by the mobile station and is used to identify redirected overload classes in global service redirection. For normal mobile stations, the 4-bit access overload class indicator is derived from the last digit of the associated decimal representation of the IMSI_M via decimal to binary conversion as specified in [5] and [14].

Identifier: '6F2C'	Structure: transparent	Mandatory
SFI: '03'		
File size: 1 byte	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Access overload class ($ACCOLC_p$)	M	1 byte

Coding:

Byte 1:



5.2.13 EF_{TERM} (Call Termination Mode Preferences)

This EF contains the call termination preference MOB_TERM_HOMEp, MOB_TERM_SIDp and MOB_TERM_FOR_NIDp.

Identifier: '6F2D'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

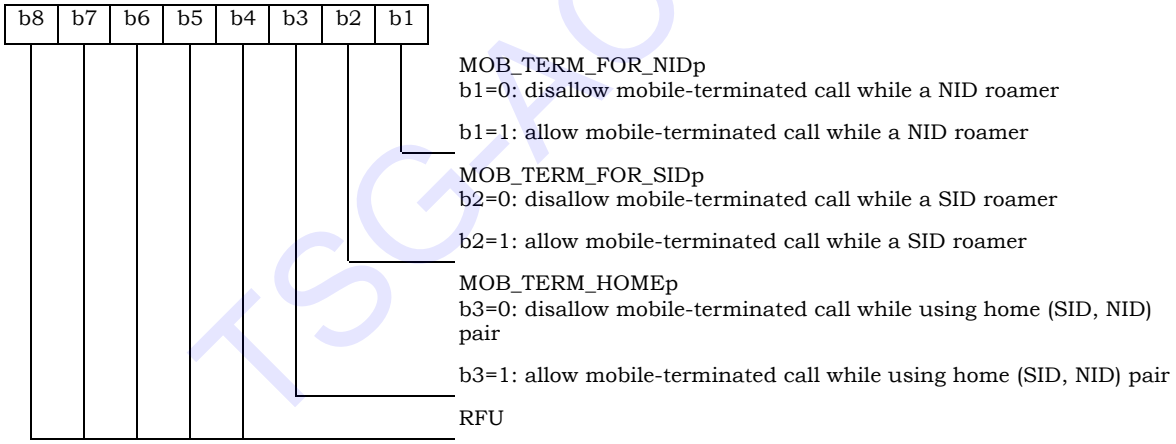
Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Call termination preferences	M	1 byte

Coding:

Byte 1:



5.2.14 EF_{SSCI} (Suggested Slot Cycle Index)

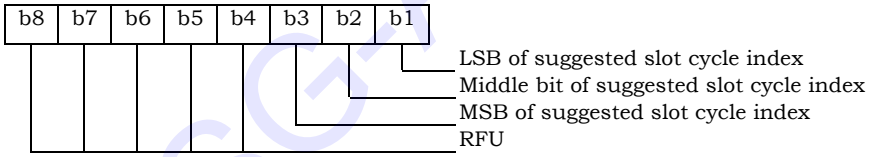
This EF suggests a value for the mobile station's preferred slot cycle index for CDMA operation (see Section 2.3.11 of [5] or 6.3.11 of [14]). Since the mobile equipment may not support all the slot cycle indexes, the mobile equipment shall select the minimum, as the preferred slot cycle index defined in [5], between the slot cycle index supported by the mobile equipment and the suggested slot cycle index contained in the EF_{SSCI}.

Identifier: '6F2E'	Structure: transparent	Optional
File size: 1 byte	Update activity: low	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Suggested slot cycle index	M	1 byte

Coding:

Byte 1:



5.2.15 EF_{ACP} (Analog Channel Preferences)

This EF specifies the analog mode channel preferences as determined by the service provider in accordance with the terms of the subscription. The items addressed are the Analog Initial Paging Channel, the Analog First Dedicated Control Channel for System A, the Analog First Dedicated Control Channel for System B, and the Number of Dedicated Control Channels to scan.

Identifier: '6F2F'	Structure: transparent	~~Mandatory~~Optional
File size: 7 bytes	Update activity: low	

Access Conditions:

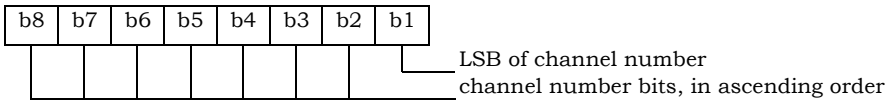
READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1-2	Analog Initial Paging Channel	M	2 bytes
3-4	Analog First Dedicated Control Channel System A	M	2 bytes
5-6	Analog First Dedicated Control Channel System B	M	2 bytes
7	Number of Dedicated Control Channel to Scan	M	1 byte

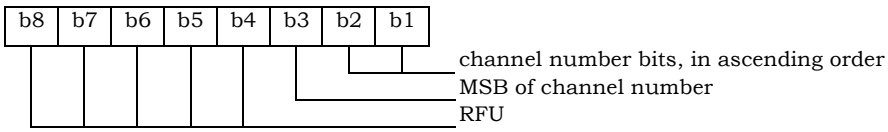
NOTE: Each channel is represented by an 11-bit binary number.

Coding:

Byte 1, 3, 5:



Byte 2, 4, 6:



5.2.16 EF_{PRL} (Preferred Roaming List)

This EF stores the Preferred Roaming List, as described in Section ~~3.5.3~~3.5.5 of [7].

Identifier: '6F30'	Structure: transparent	Mandatory
SFI: '07'		
File size: ~~'~~MAX_PR_LIST_SIZE~~'~~ for EFPRL	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1-PR_LIST_SIZE	PR_LIST ~~(see Section 3.5.5 of [7])~~	M	PR_LIST_SIZE

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

- PR_LIST

  Contents:

  The Preferred Roaming List.

  Coding:

  As defined in section 3.5.5 of [7].

5.2.17 EF_{RUIMID} (Removable UIM ID)

This EF stores a 32-bit electronic identification number (ID) unique to the CSIM or a 32-bit pseudo pUIMID of the CSIM. The file may store a 32-bit pseudo pUIMID constructed in the following way: — — The most significant 8 bits shall be 0x80. The and the least significant 24 bits shall be the 24 least significant bits of SHA-1 digest of the entire E UIMIDEUIMID, either LF_EUIMID or SF_EUIMID[2] (based on service n34 in EF_{CSIM_ST}) [3].

Identifier: '6F31'	Structure: transparent	Mandatory
File size: 5 or 8 bytes	Update activity: low	
Access Conditions: READ ALW UPDATE Never INVALIDATE Never REHABILITATE Never		

Bytes	Description	M/O	Length
1	Number of bytes	M	1 byte
2	Lowest-order byte	M	1 byte
3	:	M	1 byte
4	:	M	1 byte
5	:	M	1 byte
6	:	O	1 byte
7	:	O	1 byte
8	Highest-order byte	O	1 byte

[2] Example: if the LF_EUIMID (ICCID) is (hexadecimal) 89 (MSB) 01 01 01 23 45 67 89 01 4F (LSB), the pseudo-UIMID is (hexadecimal) 80 (Byte 5) C5 D5 64 (Byte 2), and with Byte 1 set to 04; if the 56-bit SF_EUIMID is (hexadecimal) FF (MSB) 00 00 01 12 34 56 (LSB), the pseudo-UIMID is (hexadecimal) 80(Byte 5) 07 37 E1(Byte 2), and with Byte 1 set to 04.

[3]The EUIMID (either form) is loaded into a 512-bit SHA-1 input block, starting with bit 1 of this block, to produce an output, from which the least significant 24 bits are used as the least significant 24 bits of EF(RUIMID). The 4-bit digits of EUIMID are loaded in the order d1, d2, d3, d4...dn-1, dn. Numbering the SHA-1 input buffer bits from 1 (first loaded) upwards, for each digit the most significant bit is loaded into the lowest numbered of four consecutive SHA-1 input bits and the least significant bit into the highest.

5.2.18 EF_{CSIM_ST} (CSIM Service Table)

This EF indicates which services are available~~,~~. If a service is ~~not~~ indicated as not available in the CSIM, the ME shall not select or use ~~this~~ that service.

Identifier: '6F32'	Structure: transparent	Mandatory
SFI: '02'		
File size: X bytes, X>=1	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Services n1 to n8	M	1 byte
2	Services n9 to n16	O	1 byte
3	Services n17 to n24	O	1 byte
4	Services n25 to n32	O	1 byte
:	:	:	:
X	Services n(8X-7) to n(8X)	O	1 byte

Services:		
	Service n1 :	Local Phone book
	Service n2 :	Fixed Dialing Numbers (FDN)
	Service n3 :	Extension 2
	Service n4 :	Service Dialing Numbers (SDN)
	Service n5 :	Extension 3
	Service n6 :	Short Message Storage ~~(SMS)~~
	Service n7 :	Short Message Parameters
	Service n8 :	HRPD
	Service n9 :	Service Category Program for BC-SMS
	Service n10 :	CDMA Home Service Provider Name
	Service n11 :	~~Data Download~~ via SMS Broadcast (for CCAT)
	Service n12 :	Data Download via SMS-PP (for CCAT)
	Service n13 :	Call Control for Voice Services

Services:		
	Service n14 :	3GPD-SIP
	Service n15 :	3GPD-MIP
	Service n16 :	AKA
	Service n17 :	IP-based Location Services (LCS)
	Service n18 :	BCMCS
	Service n19 :	Multimedia Messaging Service (MMS)
	Service n20 :	Extension 8
	Service n21 :	MMS User Connectivity Parameters
	Service n22 :	Application Authentication
	Service n23 :	Group Identifier Level 1
	Service n24 :	Group Identifier Level 2
	Service n25 :	De-Personalization Control Keys
	Service n26 :	Cooperative Network List
	Service n27 :	Outgoing Call Information (OCI)
	Service n28 :	Incoming Call Information (ICI)
	Service n29 :	Extension 5
	Service n30 :	Multimedia Storage
	Service n31 :	Image (EF_{IMG})
	Service n32:	~~Enabled Services Table~~Reserved[4]
	Service n33:	Reserved[5]~~Capability Configuration Parameters (CCP)~~
	Service n34:	SF_EUIMID-based EUIMID
	Service n35:	Messaging and 3GPD Extensions
	Service n36:	Root Certificates
	Service n37:	WAP Browser
	Service n38:	Java
	Service n39:	Reserved for CDG

[4] The Enabled Services Table service is deprecated. Some compliant CSIMs may have this service available.

[5] The Capability Configuration Parameters (CCP) service is deprecated. Some compliant CSIMs may have this service available.

Services:		
	Service n40:	Reserved for CDG
	Service n41:	IPv6
	Service n42:	Call Control for Data Services
	Service n43:	Call Control for Mobile Originated SMS
	Service n44	Card supports ENVELOPE (USSD Data Download) and ENVELOPE(Call Control) for USSD

The EF shall contain at least one byte. Further bytes may be included, but if the EF includes an optional byte, then it is mandatory for the EF to also contain all bytes before that byte. Other services are possible in the future and will be coded on further bytes in the EF. The coding falls under the responsibility of the 3GPP2.

Coding:

1 bit is used to code each service:

bit = 1: service available;

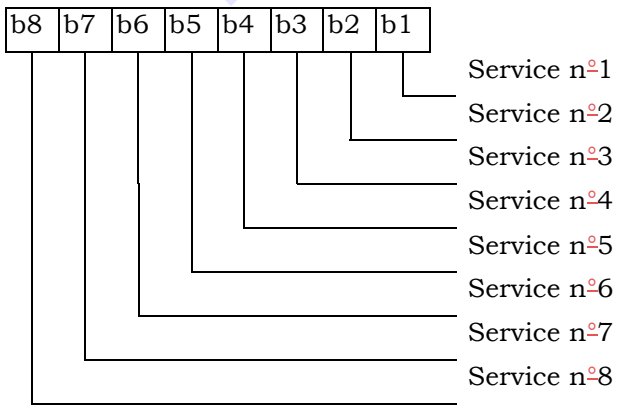
- The CSIM has the capability to support the service and that the service is available for the user of the CSIM unless the service is identified as "disabled" in EF_{EST}.

bit = 0: service not available.

- The service shall not be used by the CSIM user, even if the CSIM has the capability to support the service.

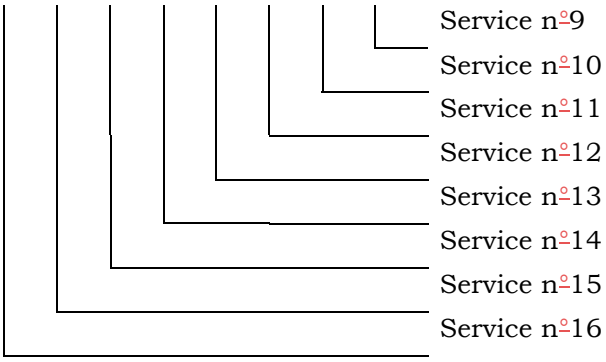
- ~~Service available means that the CSIM has the capability to support the service and that the service is available for the user of the CSIM unless the service is identified as "disabled" in EF_{EST}. Service not available means that the service shall not be used by the CSIM user, even if the CSIM has the capability to support the service.~~

First byte:



Second byte:

b8	b7	b6	b5	b4	b3	b2	b1



etc.

If the FDN feature is enabled according to EF_{EST}, destination addresses which are not stored in EF_{FDN} shall not be allowed on any voice call or SMS.

~~If the CSIM supports the FDN feature (FDN is enabled in EF_{EST}) a special mechanism shall exist in the CSIM which invalidates **EF_{IMSI_T}**, **EF_{IMSI_M}** and **EF_{TMSI}** once during each CDMA session. This mechanism shall be invoked by the CSIM automatically if FDN is enabled. This invalidation shall occur at least before the next command following selection of either **EF_{FDN}** is enabled when the ADN is invalidated or not available.~~

If service n34 (SF_EUIMID-based EUIMID) is not available, ME shall fill in EXT_UIM_ID INFO RECORD with the entire contents of EF_{ICCID} in response to Status Request Message defined in [5]. Otherwise, ME shall fill in EXT_UIM_ID INFO RECORD with SF_EUIMID from EF_{SF_EUIMID}

5.2.19 EF_{SPC} (Service Programming Code)

This EF includes the Service Programming Code (SPC), having a value from 0 to 999,999. The default value is 0. Details of SPC are in [7] Section 3.3.6.

Identifier: '6F33'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: low	

Access Conditions:

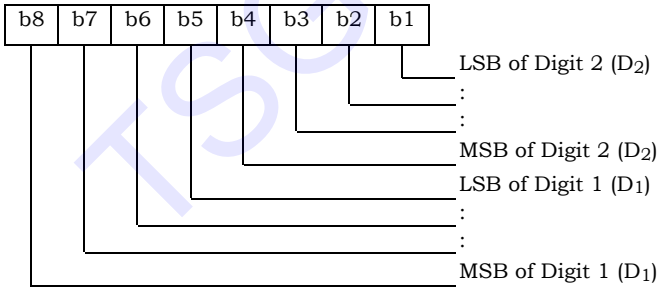
READ	ADM
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1-3	Service Programming Code	M	3 bytes

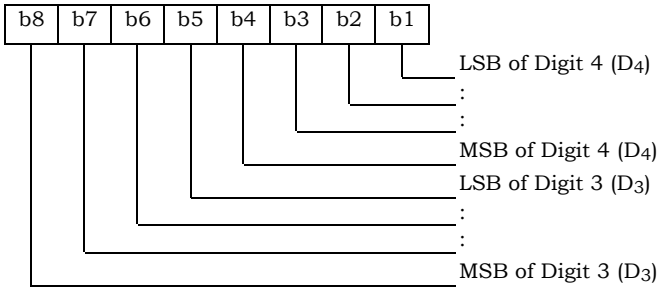
Coding:

SPC is a 6-digit number$D_1D_2D_3D_4D_5D_6$, where D_1 is the most significant digit and D_6 is the least significant digit. The coding of SPC in this EF is according to [7], Section 4.5.4.2, whereby each digit is encoded in BCD format.

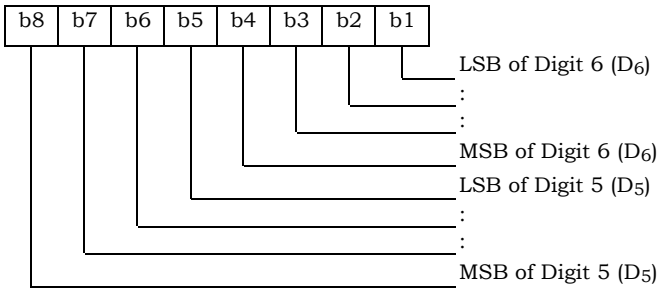
Byte 1:



Byte 2:



Byte 3:



5.2.20 EF$_{OTAPASPC}$ (OTAPA/SPC_Enabled)

This EF contains user-entered control information that either prevents or (else) permits network manipulation of the SPC, and either prevents or (else) permits OTAPA to be performed on the NAM. This EF is based upon information in [7], Sections 3.2.2 and 3.3.6. A successful base station response to an CSIM initiated challenge is required prior to any network manipulation of OTAPA accessible files.

Identifier: '6F34'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

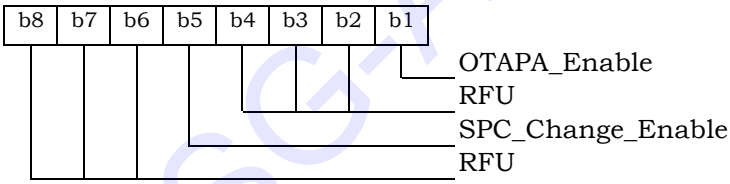
Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	OTAPA/SPC_Enable	M	1 byte

Coding:

Byte 1:



For "OTAPA_Enable", a value of '0' for the NAM indicates that the user consents to the performance of OTAPA for the NAM by the service provider. A value of '1' indicates that the user does not permit OTAPA to be performed on the NAM. Refer to [7], Section 3.2.2.

For "SPC_Change_Enable", a value of '0' for the CSIM indicates that the user consents to allow the service provider to change ~~the value of~~ the Service Programming Code from a default value (zero) to a non-default value (non-zero). A value of '1' indicates that the user denies permission for the service provider to change the ~~value of~~ SPC from a default value to a non-default value. See Sec. 3.3.6 of [7].

5.2.21 $EF_{NAMLOCK}$ (NAM_LOCK)

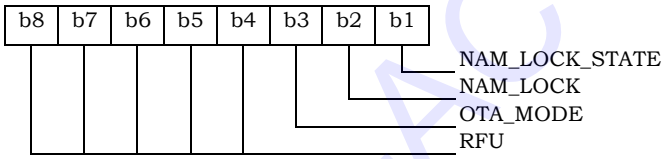
This EF stores the locked/unlocked state of the NAM. This EF is based upon information in [7].

Identifier: '6F35'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	SPASM protection indicator (NAM_LOCK) status	M	1 byte

Coding:

Byte 1:



Although the UPDATE Access Condition for this EF is PIN, the ME cannot update bits 1 and 2. If the ME sends an UPDATE BINARY command for this EF, the CSIM shall only use b3 through b8 and shall ignore bits b1 and b2.

Bit 1 gives the current NAM_LOCK_STATE. A value of '1' indicates that the NAM is locked by the SPASM protection mechanism. A value of '0' indicates that the NAM is unlocked.

Bit 2 gives the permanent NAM_LOCK setting (equivalent to $NAM\_LOCK_{S\text{-}P}$ in [7]). A value of '1' indicates that the SPASM protection mechanism shall ~~must~~ be satisfied for network initiated OTA. A value of '0' indicates that SPASM protection is not required.

Bit 3 gives the OTA_MODE for the current OTA session. A value of '0' indicates user-initiated, and a value of '1' indicates network-initiated.

If an OTA programming session was initiated by the user as described in Section 3.2.1 of [7] (i.e. OTA_MODE in $EF_{NAMLOCK}$ is '0'), SPASM does not protect access to the NAM parameters and indicators. In this case, the CSIM ~~ME~~ shall set the NAM_LOCK_STATE (b1) to '0~~.~~' ~~The~~ and NAM_LOCK (b2) ~~bit~~ shall not be changed.

On invocation of a network-initiated OTA session (after receiving an OTAPA REQUEST), the ~~ME~~ CSIM shall set the value of the NAM_LOCK_STATE (b1) to the value of NAM_LOCK (b2)~~NAM_LOCK_STATE=NAM_LOCK~~. The conditions under which the values of NAM_LOCK_STATE and NAM_LOCK should be changed are defined in sections 3.3.1.10 (Validation Request Message) and 3.3.1.3 (Commit Request Message) of [7].

The ME updates the OTA_MODE bit to tell the CSIM how an OTA session was initiated. The ME shall set this bit on initiation of an OTA session. The CSIM shall comply with the requirements in [7] (e.g. shall reject ~~OTAPA Request~~OTAPA REQUEST while in a user-initiated session~~.~~).

5.2.22 EF_{OTA} (OTASP/OTAPA Features)

This EF stores a listing of OTASP/OTAPA features supported by the CSIM, along with protocol revision codes. This EF is based on the format and coding rules in Section 3.5.1.7 of [7], including the subset of fields described below. ~~This EF is a subset of the information in [7], Section 3.5.1.7.~~

Identifier: '6F36'	Structure: transparent	Mandatory
File size: 2*NUM_FEATURES + 1 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	NUM_FEATURES, number of OTASP/OTAPA features	M	1 byte
2	First FEATURE_ID	M	1 byte
3	First FEATURE_P_REV	M	1 byte
	...		
2*NUM_FEATURES	Last FEATURE_ID	M	1 byte
2*NUM_FEATURES + 1	Last FEATURE_P_REV	M	1 byte

Identifier: '6F36'	Structure: transparent	Mandatory
File size: 2N + 1 bytes	Update activity: low	

Access Conditions:

READ PIN

UPDATE ADM

INVALIDATE ADM

REHABILITATE ADM

Bytes	Description	M/O	Length
1	N, number of OTASP/OTAPA features	M	1 byte
2	NAM Download (DATA_P_REV) ID	M	1 byte
3	DATA_P_REV	M	1 byte
4	Key Exchange (A_KEY_P_REV) ID	M	1 byte
5	A_KEY_P_REV	M	1 byte
6	System Selection for Preferred Roaming (SSPR_P_REV) ID	M	1 byte
7	SSPR_P_REV	M	1 byte
8	Service Programming Lock (SPL_P_REV) ID	M	1 byte
9	SPL_P_REV	M	1 byte
10	Over The Air Parameter Admin (OTAPA_P_REV) ID	M	1 byte
11	OTAPA_P_REV	M	1 byte
12	Preferred User Zone List (PUZL_P_REV) ID	M	1 byte
13	PUZL_P_REV	M	1 byte
14	3G Packet Data (3GPD) ID	M	1 byte
15	3GPD	M	1 byte
16	Secure MODE (SECURE_MODE_P_REV) ID	M	1 byte
17	SECURE_MODE_P_REV	M	1 byte
÷	÷	÷	÷
2N	Feature N	M	1 byte
2N + 1	Protocol Revision for Feature N	M	1 byte

NOT C.S0016-A v3.0

NOTE: Coding of features (FEATURE_ID) and protocol revisions (FEATURE_P_REV) ~~are~~ is described in Table 3.5.1.7-1 (Feature Identifier) of [7]~~, Section 3.5.1.7~~.

5.2.23 EF_{SP} (Service Preferences)

This EF describes the user's service preferences as defined in ~~[14]~~Section 2.3.10.1 of [5] or Sections 6.3.10.1 and 6.3.10.2 of [14].

Identifier: '6F37'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

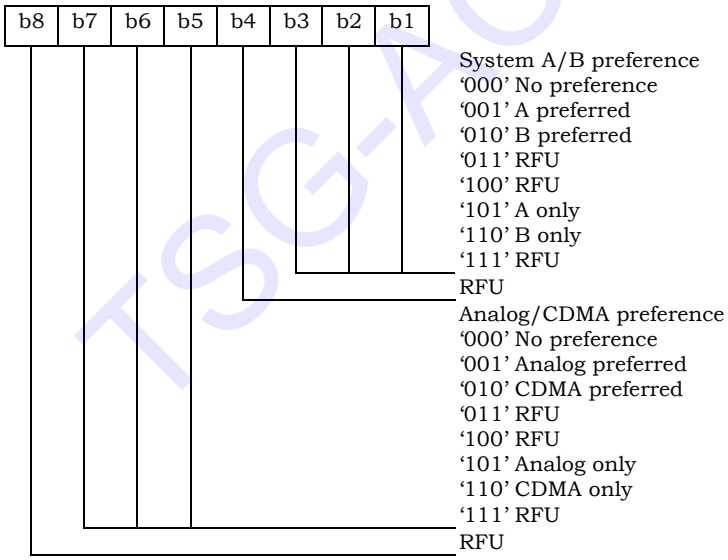
Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Service Preferences (e.g. band class, analog vs. CDMA)	M	1 byte

Coding:

Byte 1:



5.2.24 EF$_{ESN_MEID_ME}$ (ESN_ME or MEID_ME)

This EF stores the ~~32-bit Electronic Serial Number~~ESN_ME or ~~56-bit MEID~~MEID_ME ~~or 32-bit pseudo-ESN~~ of the Mobile Equipment (ME) to which the CSIM is attached. ~~This number is transferred to the CSIM when the ME determines that the CSIM has been inserted~~during ~~initialization.~~

Identifier: '6F38'	Structure: transparent	Mandatory
File size: 8 bytes	Update activity: low	
Access Conditions: READ ALW UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Number of bytes for ESN_ME or MEID_ME	M	1 byte
2	~~Lowest-order byte~~Least significant byte	M	1 byte
3	:	M	1 byte
4	:	M	1 byte
5	:	M	1 byte
6	:	M	1 byte
7	:	M	1 byte
8	~~Highest-order byte~~Most significant byte	M	1 byte

Unused bytes shall be set to '00'.

5.2.25 Reserved

5.2.26 EF_{LI} (Language Indication)

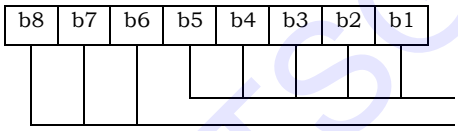
This EF contains the codes for one or more languages. This information, determined by the user/operator, defines the preferred languages of the user in order of priority. This information may be used by the ME for ~~MMI~~ UI purposes.

Identifier: '6F3A'	Structure: transparent	Optional
SFI: '0A'		
File size: 2N bytes	Update activity: low	
Access Conditions: READ ALW UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 – 2	1st language code (highest priority)	M	2 bytes
3 – 4	2nd language code	O	2 bytes
:	:	:	:
2N-1 – 2N	Nth language code (lowest priority)	O	2 bytes

Coding:

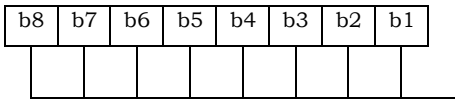
Byte 1:



~~Character~~ CHARi Encoding type as ~~defined~~ shown in Table 9.1-1 , "Data Field Encoding Assignments", in [Informative 1].

RFU

Byte 2:



Language Indicator as ~~defined~~ shown in Table 9.2-1, "Language Indicator Value Assignments", in [Informative 1].

5.2.27 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 at the CSIM ADF level. It may also contain an associated alpha-tagging. If this file is present in the CSIM, the Enabled Services Table (EF_{EST}) shall also be present.

Identifier: '6F3B'	Structure: linear fixed	Optional
Record length: X+14 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE PIN2 DEACTIVATE ADM ACTIVATE 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	Reserved ('FF')~~Capability/Configuration2 (EF_{CCP2}) Record Identifier~~	M	1 byte
X+14	Extension2 (EF_{EXT2}) Record Identifier	M	1 byte

For contents and coding of all data items, see the respective data items of the EF_{ADN} (Section 5.4.1), 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}.

5.2.28 EF_{SMS} (Short Messages)

This EF contains information in accordance with [8] comprising short messages (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: '6F3C'	Structure: linear fixed	Optional
Record Length: variable (1)	Update activity: high	

Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Status	M	1 byte
2	MSG_LEN	M	1 byte
3 – 3+MSG_LEN	SMS Transport Layer Message	M	MSG_LEN bytes

Note: (1) The length and the byte allocations are variable according to the actual size of the SMS Transport Layer message. The maximum length is 255, which includes the length of the short message plus two bytes for storing "status" and "MSG_LEN".

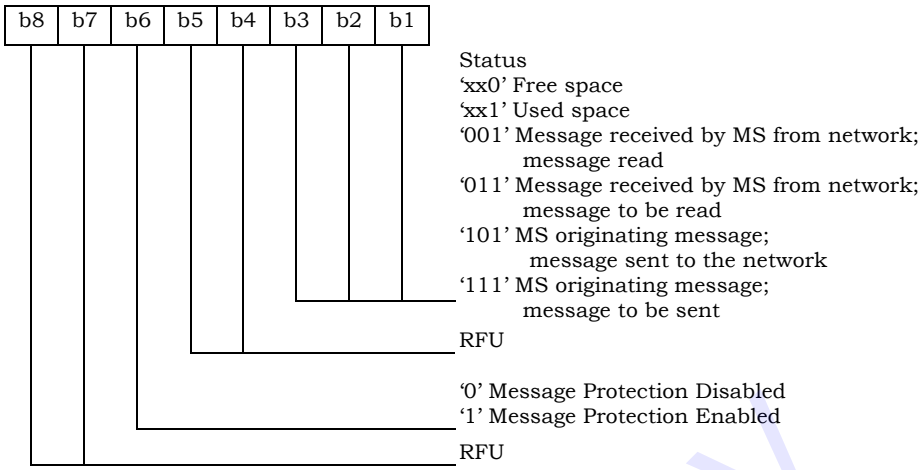
- Status

  Contents:

  Status byte of the record which can be used as a pattern in the SEEK command. For MS originating messages sent to the network, the status shall be updated when the MS receives a status report or sends a successful SMS Command relating to the status report.

Coding:

Byte 1:



- MSG_LEN

Contents:

The length of the message, not including MSG_LEN. Note that the definition of this EF does allow multiple occurrences of the segment, which consists of "PARAMETER_ID", "PARAMETER_LEN", and "Parameter Data" as described in [8]. The number of repetitions of the aforementioned segment is determined by MSG_LEN and the PARAMETER_LEN of each segment.

- SMS Transport Layer Message

Contents: see Section 3.4.1 of [8].

5.2.29 EF_{SMSP} (Short Message Service Parameters)

This EF contains values for Short Message Service ~~header~~ Parameters (SMSP), which can be used by the Mobile Equipment (ME) for user assistance in preparation of mobile originated short messages.

The EF consists of one or more records, with each record able to hold a set of SMS parameters. The first (or only) record in the EF shall be used as a default set of parameters, if no other record is selected. To distinguish between records, a four-byte Teleservice Identifier as defined in [8] shall be included within each record. The SMS parameters stored within a record may be present or absent independently. When an SMS~~a short message~~ is to be sent ~~from the Mobile Station (MS)~~, the parameters in the CSIM record that has the same Teleservice Identifier as the one in the mobile-originated message, if present, ~~shall~~ can be used by the ME when a value is not supplied by the user.

Identifier: '6F3D'	Structure: linear fixed	Optional
Record Length: variable (10+X)	Update activity: high	

Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
~~(1), (2)~~1-4	Teleservice Identifier	M	4 bytes
5-6	Parameter Indicators	M	2 bytes
7	Reserved	M	1 byte
8-(8+N-1)	Destination Address	M	~~Variable (1)(3)~~N (Note 1)
8+N	MSG_ENCODING	M	1 byte
9+N	Validity Period	M	1 byte
(Note 2)	Service Category	O	4 bytes
(Note 2)	Destination Subaddress	O	Variable ~~(1)~~(Note 2)
(Note 2)	Bearer Reply Option	O	3 bytes
(Note 2)	Bearer Data	O	Variable (Note 2)~~(1)~~
(Note 2) (Note 3)	Padding	O	Variable (Note 2) (Note 3)

Notes: (1) N is 1 if the Parameter Indicators field indicates that the Destination Address is absent. Otherwise, N is the length of a valid destination address. ~~See [8]~~.

(2) Starting and ending bytes and length depend on the presence and absence of parameters indicated by the Parameter Indicators field. ~~Starting and ending bytes depend on (1)~~

(3) Padding is mandatory if the fields before it do not occupy all the 10+X bytes. Padding, if present, always ends at byte number 10+X. ~~If the Destination Address is absent, the parameter length is 1 byte~~.

~~Storage is allocated for all of the possible SMS 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'.

- Teleservice Identifier

  Contents:

  The supported teleservices include ~~[16]~~*IS-91 Extended Protocol Enhanced Services*, *Wireless Paging Teleservice*, *Wireless Messaging Teleservice*, *Voice Mail Notification* and *Wireless Application Protocol*. See section 3.4.3.1 of [8] for details.

  Coding:

  4-byte Teleservice Identifier as defined in 3.4.3.1 of [8].

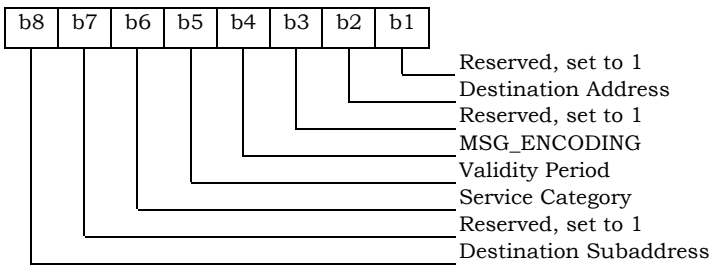
- Parameter Indicators

  Contents:

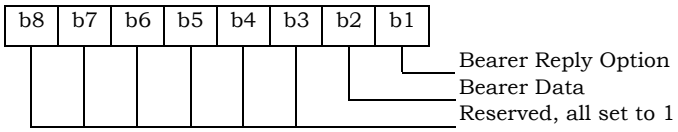
  Each of the default SMS parameters which can be stored in the remainder of the record are marked absent or present by individual bits within this byte.

  Coding:

  Byte 5:



Byte 6:



Note: Bit value 0 means parameter present
Bit value 1 means parameter absent

- Reserved

Set to 'FF'.

- Destination Address

If the Parameter Indicators field indicates this parameter is present, the contents and coding are defined in section 3.4.3.3 Address Parameters of [8]. It contains PARAMETER_ID, PARAMETER_LEN and parameter data.

If the Parameter Indicators field indicates this parameter is absent, then it shall be set to 'FF' with a length of 1 byte.

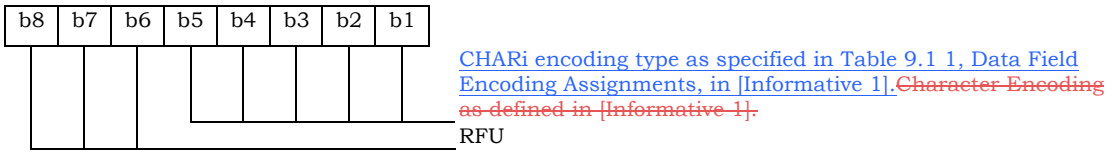
~~Contents and Coding: As defined in [8]. If this parameter is absent, then it shall be set to 'FF' with a length of 1 byte.~~

- MSG_ENCODING

Contents:

If the Parameter Indicators field indicates this parameter is present, the contents and coding are defined in Table 9.1-1 Data Field Encoding Assignments of ~~As defined in~~ [Informative 1]. This parameter can appear in the Bearer Data if Bearer Data is present. If this parameter appears in the Bearer Data too, then the same value shall be ~~set to~~used by this parameter~~; otherwise the record is invalid~~. If this parameter appears in the Bearer Data, then this parameter shall be present~~; otherwise the record is invalid~~.

Coding:



If the Parameter Indicators field indicates this field is absent, it shall be set to 'FF'.

- Validity Period

Contents and Coding:

If the Parameter Indicators field indicates this parameter is present, the contents and coding are defined in section 4.5.6 of [8] for the VALIDITY field of the ~~As defined in [8] for~~ relative time format. This parameter can appear in the Bearer Data if Bearer Data is present. If this parameter appears in the Bearer Data too, then the same value shall be ~~set to~~ used by this parameter~~; otherwise the record is invalid~~. If this parameter appears in the Bearer Data, then this parameter shall be present~~; otherwise the record is invalid~~.

If the Parameter Indicators field indicates this field is absent, it shall be set to 'FF'.

- Service Category

Contents and Coding:~~ as defined in [8].~~

As defined in section 3.4.3.2 Service Category of [8]. It contains PARAMETER_ID, PARAMETER_LEN and parameter data.

- Destination Subaddress

Contents and Coding:~~ as defined in [8].~~

As defined in section 3.4.3.4 Subaddress of [8]. It contains PARAMETER_ID, PARAMETER_LEN and parameter data.

- Bearer Reply Option

Contents and Coding:~~ as defined in [8].~~

As defined in section 3.4.3.5 Bearer Reply Option of [8]. It contains PARAMETER_ID, PARAMETER_LEN and parameter data.

- Bearer Data

Contents and Coding:~~ as defined in [8].~~

As defined in section 3.4.3.7 Bearer Data of [8]. It contains PARAMETER_ID, PARAMETER_LEN and parameter data.

- Padding

Contents and Coding:

All bytes for this field shall be set to ‘FF’.

5.2.30 EF_{SMSS} (SMS Status)

This EF contains status information relating to the short message service.

The provision of this EF is associated with EF_{SMS}. Both files shall be present together or both shall be absent from the CSIM.

Identifier: '6F3E'	Structure: transparent	Optional
File size: 5 + X bytes	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 – 2	MESSAGE_ID	M	2 bytes
3 – 4	WAP MESSAGE_ID	M	2 bytes
5	SMS "Memory Cap. Exceeded" Notification Flag/SMS Timestamp Mode	M	1 byte
6-5 + X	Reserved	O	X bytes

- MESSAGE_ID

  Contents:

  The value of the MESSAGE_ID in the last sent *SMS Submit Message* from a teleservice which requires message identifiers other than the WAP teleservice.

  Coding: as defined in [8].

- WAP MESSAGE_ID

  Contents:

  The value of the MESSAGE_ID in the last sent *SMS Submit Message* from the WAP teleservice.

  Coding: as defined in [8].

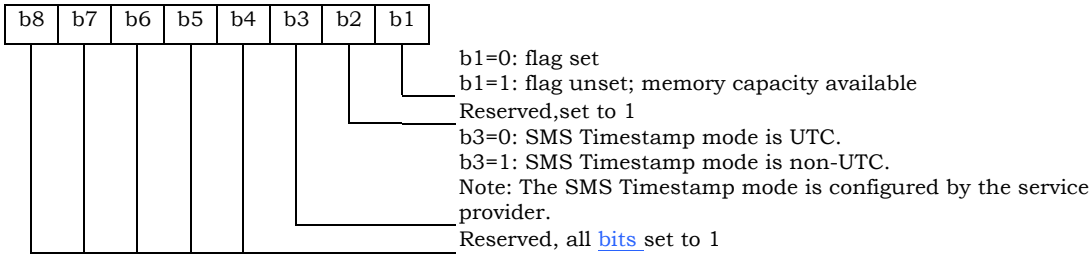
- SMS "Memory Capacity Exceeded" Notification Flag/SMS Timestamp Mode.

  Contents:

  Includes a flag that indicates whether or not there is memory capacity available to store SMS messages. Also includes a bit that indicates whether the SMS Timestamp mode is UTC or non-UTC.

Coding:

Byte 5:



5.2.31 EF_{SSFC} (Supplementary Services Feature Code Table)

This EF stores the numeric feature code to be used by the ME when a supplementary service is invoked in CDMA or analog mode via an implementation-~~dependant~~ dependent user interface (such as a menu) that automatically inserts a feature code into the dialed digit string. Because feature codes are service-provider specific, this EF is required to enable the ME to perform the mapping to the feature code.

When a supplementary service is invoked in CDMA or analog mode, the mobile station shall determine the feature code by reading the Supplementary Service Feature Code Table entry for the selected supplementary service, and pre-pending with an asterisk.

Identifier: '6F3F'	Structure: transparent	Optional
File size: 2N+1	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	N, Number of Feature Codes	M	1 byte
2 – 3	Activate Call Delivery (CD)	M	2 bytes
4 – 5	De-activate Call Delivery (CD)	M	2 bytes
6 – 7	Register new Call Forwarding – Busy (CFB) forward-to number	M	2 bytes
8 – 9	Register Call Forwarding – Busy (CFB) to voice mail	M	2 bytes
10 – 11	De-register Call Forwarding – Busy (CFB)	M	2 bytes
12 – 13	Activate Call Forwarding – Busy (CFB)	M	2 bytes
14 – 15	De-activate Call Forwarding – Busy (CFB)	M	2 bytes
16 – 17	Register new Call Forwarding – Default (CFD) forward-to number	M	2 bytes
18 – 19	Register Call Forwarding – Default (CFD) to voice mail	M	2 bytes
20 – 21	De-register Call Forwarding – Default (CFD)	M	2 bytes
22 – 23	Activate Call Forwarding – Default (CFD)	M	2 bytes
24 – 25	De- activate Call Forwarding – Default (CFD)	M	2 bytes
26 – 27	Register new Call Forwarding – No Answer (CFNA) forward-to number	M	2 bytes
28 – 29	Register Call Forwarding – No Answer (CFNA) to voice mail	M	2 bytes
30 – 31	De-register Call Forwarding – No Answer (CFNA)	M	2 bytes
32 – 33	Activate Call Forwarding – No Answer (CFNA)	M	2 bytes
34 – 35	De-activate Call Forwarding – No Answer (CFNA)	M	2 bytes

Bytes	Description	M/O	Length
36 – 37	Register new Call Forwarding – Unconditional (CFU) forward-to number	M	2 bytes
38 – 39	Register Call Forwarding – Unconditional (CFU) to voice mail	M	2 bytes
40 – 41	De-register Call Forwarding – Unconditional (CFU)	M	2 bytes
42 – 43	Activate Call Forwarding – Unconditional (CFU)	M	2 bytes
44 – 45	De-activate Call Forwarding – Unconditional (CFU)	M	2 bytes
46 – 47	Activate Call Waiting (CW)	M	2 bytes
48 – 49	De-activate Call Waiting (CW)	M	2 bytes
50 – 51	Temporarily De-activate Call Waiting (Cancel Call Waiting - CCW)	M	2 bytes
52 – 53	Temporarily Activate Calling Number Identification Restriction (CNIR) (per-call blocking)	M	2 bytes
54 – 55	Temporarily De-activate Calling Number Identification Restriction (CNIR) (per-call allowed)	M	2 bytes
56 – 57	Invoke Conference Calling (CC)	M	2 bytes
58 – 59	Invoke Drop Last Conference Calling (CC) Party	M	2 bytes
60 – 61	Activate Do Not Disturb (DND)	M	2 bytes
62 – 63	De-activate Do Not Disturb (DND)	M	2 bytes
64 – 65	Activate Message Waiting Notification (MWN) Alert Pip Tone	M	2 bytes
66 – 67	De-activate Message Waiting Notification (MWN) Alert Pip Tone	M	2 bytes
68 – 69	Activate Message Waiting Notification (MWN) Pip Tone	M	2 bytes
70 – 71	De-activate Message Waiting Notification (MWN) Pip Tone	M	2 bytes
72 – 73	Temporarily De-activate Message Waiting Notification (MWN) Pip Tone (Cancel MWN - CMWN)	M	2 bytes
74 – 75	Invoke Priority Access and Channel Assignment (PACA)	M	2 bytes
76 – 77	Invoke Voice Message Retrieval (VMR)	M	2 bytes
78 – 79	Activate Calling Name Presentation (CNAP)	M	2 bytes
80 – 81	De-activate Calling Name Presentation (CNAP)	M	2 bytes
82 – 83	Activate Calling Name Restriction (CNAR)	M	2 bytes
84 – 85	De-activate Calling Name Restriction (CNAR)	M	2 bytes
86 – 87	Activate Automatic Callback (AC)	M	2 bytes
88 – 89	De-activate Automatic Callback (AC)	M	2 bytes
90 – 91	Activate Automatic Recall (AR)	M	2 bytes
92 – 93	De-activate Automatic Recall (AR)	M	2 bytes
94 – 95	Register new network registered User Selectable Call Forwarding (USCF) directory number	M	2 bytes
96 – 97	Activate Rejection of Undesired Annoying Calls (RUAC)	M	2 bytes
98 – 99	De-activate Rejection of Undesired Annoying Calls (RUAC)	M	2 bytes
100 –	Invoke Advice of Charge (AOC)	M	2 bytes

Bytes	Description	M/O	Length
101			
102 – 103	Invoke Call Trace (COT)	M	2 bytes
2N – 2N+1	FCN	M	2 bytes

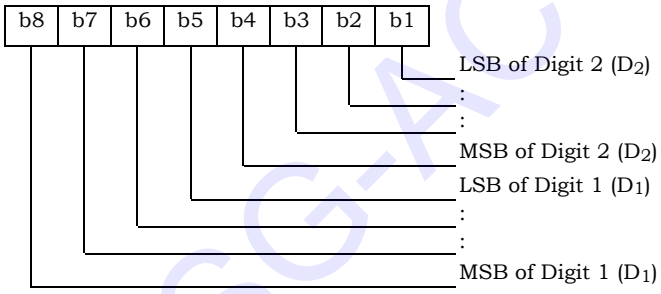
N, Number of Feature Codes" , ~~is coded in hexadecimal value, which~~ indicates the number of feature codes.

A feature code of up to four digits shall be encoded via BCD into the two bytes of the feature code table entry as follows:

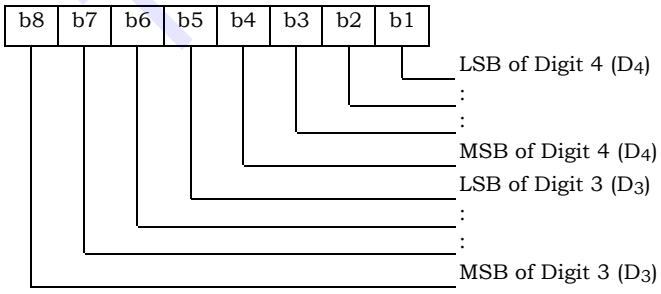
- represent these four digits as $D_1D_2D_3D_4$.
- if the feature code (FC) ~~of~~ is less than four digits ~~is used~~, ~~the digits shall be right justified and~~ the unused digits shall be set to 'F'.

Coding:

First byte:



Second byte:



5.2.32 EF_{SPN} (CDMA Home Service Provider Name)

If service n10 is available, this EF shall be present. This EF contains the home service provider name and appropriate requirements for display by the ME.

Identifier: '6F41'	Structure: transparent	Optional
SFI: '08'		
File size: 35 bytes	Update activity: low	

Access Conditions:

READ	ALW
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Display Condition	M	1 byte
2	Character Encoding	M	1 byte
3	Language Indicator	M	1 byte
4 – 35	Service Provider Name	M	32 bytes

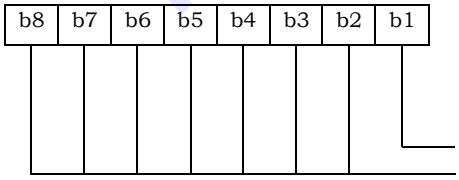
- Display Condition

  Contents:

  An indication of whether or not ~~a~~ the home service provider name ~~should~~ shall be displayed by a MS which supports this feature~~when the MS is registered in the home service area~~.

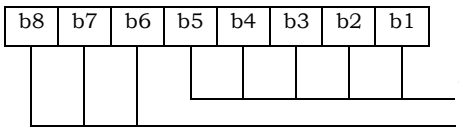
  Coding:

  Byte 1:



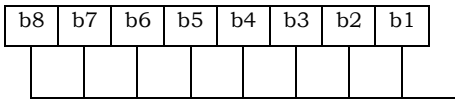
b1=0: display of ~~registered system~~home service provider name is not required
b1=1: ~~display of registered system is required~~display of home service provider name is required
RFU

  Byte 2:



~~Character~~ CHARi encoding type as specified in [Informative 1], Table 9.1-1, Data Field Encoding Assignments.
RFU

Byte 3:



Language Indicator as specified in [Informative 1], Table 9.2-1, Language Indicator Value Assignments

Bytes 4 – 35:

- Service Provider Name

  Contents: ~~service~~ Service provider string to be displayed.

~~Coding:~~

The string shall use SMS conventions as defined in Tables 9.1-1 and ~~9-2~~9.2-1 of [~~Informative~~ Informative 1]. The string shall be ~~left justified~~stored in sequence with the first character in byte 4. Unused bytes shall be stored in the highest numbered bytes and shall be set to 'FF'.

If the string is coded as 7-bit, the SMS default 7-bit coded alphabet as referenced in [Informative 1] with bit 8 set to 0 shall be used.

5.2.33 EF$_{USGIND}$ (UIM_ID/SF_EUIMID Usage Indicator)

This EF indicates whether the ~~32 bits of the~~ UIM_ID or ESN_ME is used as the ~~"~~ESN~~"~~ value for CAVE authentication and MS identification, as per Section 4.6.1 of [46]. This EF also indicates whether the ~~56 bits of the~~ SF_EUIMID or MEID_ME shall be used as the ~~"MEID"~~MEID field over the air when Service n34 is available. ~~This indicator shall be set to comply with US Code of Federal Regulations 47 (CFR) 1998 Part 22.919, where applicable.~~

Identifier: '6F42'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	UIM-ID/SF_EUIMID Usage Indicator	M	1 byte

Coding:

~~1 bit is used as the UIM ID usage indicator.~~

~~first bit = 0: ESN_ME is used for CAVE authentication and MS identification.~~

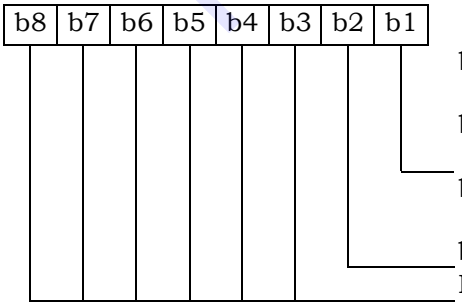
~~first bit = 1: UIM_ID is used for CAVE authentication and MS identification.~~

~~1 bit is used as the SF_EUMID usage indicator.~~

~~second bit = 0: MEID is used for MS identification.~~

~~second bit = 1: SF_EUIMID is used for MS identification~~

Byte 1:



b1=0: ~~-~~ESN_ME is used for CAVE Authentication and MS Identification.

b1=1: ~~UIM_ID~~UIMID is used for CAVE Authentication and MS Identification.

b2=0: ~~MEID~~ MEID_ME is used for MS Identification.

b2=1: ~~-~~SF_EUIMID is used for MS Identification.

RFU

~~The default value for b1 shall be set to '0'.~~

~~If service n34 is not available, the b2 bit shall be set to '0' and shall not be interpreted by the ME.~~

~~If service n34 is available and activated and the ME is assigned with ESN, then the b2 shall not be interpreted~~

The ME shall interpret b2 only if the ME is assigned with an MEID_ME and service n34 is available.

5.2.34 EF_{AD} (Administrative Data)

This EF contains information concerning the mode of operation according to the type of UIM. It also provides an indication whether some ME features should be activated during the normal operation.

Identifier: '6F43'	Structure: transparent	Mandatory
SFI: '01'		
File size: 3+X bytes	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
2 – 3	Additional information	M	2 bytes
4 – 3+X	RFU	O	X bytes

- MS operation mode

Contents:

mode of operation for the MS.

Coding:

Initial value

- normal operation '00'.

Refer to [17] for other operational values.

Byte 1:

b8	b7	b6	b5	b4	b3	b2	b1

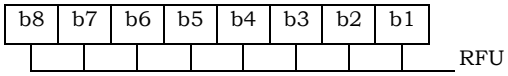
b8 through b1='~~00000000~~'00000000'.

- Additional information

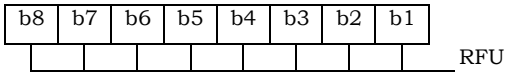
Coding:

- specific facilities (if b1=1 in byte 1);

Byte 2: (first byte of additional information)



Byte 3:



5.2.35 EF_{MDN} (Mobile Directory Number)

This EF stores the Mobile Directory Number, Type of Number, Numbering Plan, Presentation Indicator and Screening Indicator.

Identifier: '6F44'	Structure: linear fixed	Optional
Record length: 11 bytes	Update activity: low	

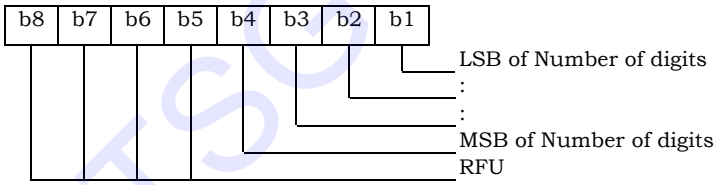
Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
~~1~~	~~RFU~~ ~~Number of digits~~	~~M~~	~~1 byte~~
1	Number of digits	M	1 byte
2 – 9	MDN	M	8 bytes
10	NUMBER_TYPE and NUMBER_PLAN	M	1 byte
11	PI and SI	M	1 byte

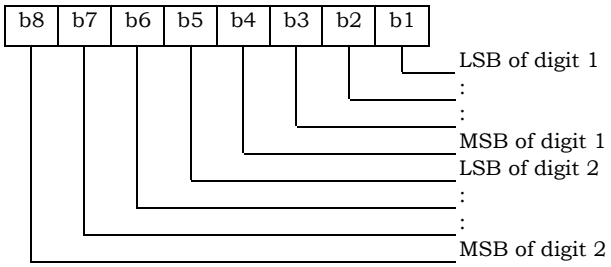
Coding:

Byte 1:

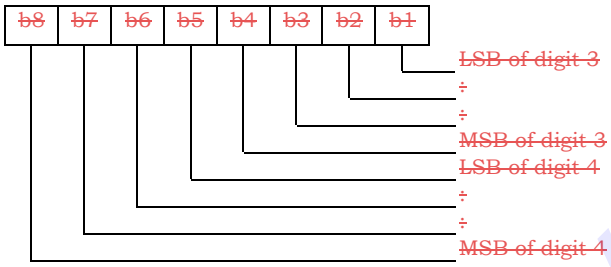


Byte 2 through 9 store MDN up to 15 digits described in Section 2.3.1.4 of [5] and Section 6.3.1.4 of [14]. Each digit shall be encoded according to Table 2.7.1.3.2.4-4 of [5] and Table 6.7.1.3.2.4-4 of [14]. If MDN requires less than 15 digits, excess nibbles at the end of data shall be set to 'F'.

Byte 2:

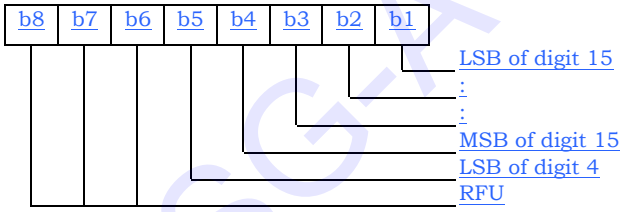


~~Byte 3:~~

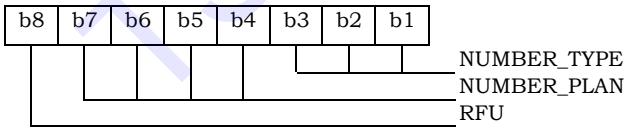


And ~~Byte 4~~Bytes 3 through ~~9~~ 8 shall follow the same format as Byte~~s~~ 2 ~~and 3~~.

Byte 9:

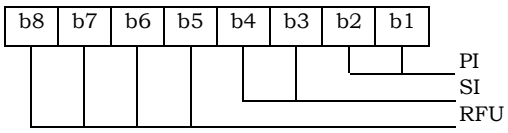


Byte 10:



Refer to Section 2.7.4.4 of [5] or ~~[14],~~ Section 6.7.4.4 of [14].

Byte 11:



Refer to Section 2.7.4.4 of [5] or ~~[14],~~ Section 6.7.4.4 of [14].

5.2.36 EF_{MAXPRL} (Maximum PRL)

This EF stores the maximum size, in octets, that the CSIM can support for EF Preferred Roaming List and EF Extended Preferred Roaming List. See 3.5.3.1 and 3.5.3.3 of [7] for more detail.

Identifier: '6F45'	Structure: transparent	Mandatory
File size: 2 or 4 bytes	Update activity: Never	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 – 2	MAX_PR_LIST_SIZE for EF_{PRL}	M	2 bytes
3 – 4	MAX_PR_LIST_SIZE for EF_{EPRL}	O	2 bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

The 'MAX_PR_LIST_SIZE for EF_{EPRL}' field shall be included if EF_{EPRL} is present.

5.2.37 EF_{SPCS} (SPC Status)

This EF identifies whether the EF_{SPC} (Service programming code) is set to default and internally updated in the card to reflect the current state of SPC after an OTASP commit if the SPC was changed. Details of SPC are in [7], Section 3.3.6.

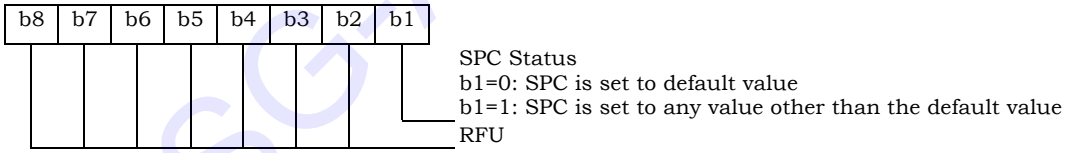
Identifier: '6F46'	Structure: transparent	Mandatory
File size: 1 byte	Update activity: low	
Access Conditions: READ PIN UPDATE ~~NEVER~~Never INVALIDATE Never~~NEVER~~ REHABILITATE Never~~NEVER~~		

Bytes	Description	M/O	Length
1	SPC Status	M	1 byte

- SPC Status

Coding:

Byte 1:



5.2.38 EF_{ECC} (Emergency Call Codes)

This EF contains ~~up to 5~~ emergency call codes (ECCs).

Identifier: '6F47'	Structure: transparent	Optional
SFI: '09'		
File size: 3n ~~(n ≤ 5)~~ bytes	Update activity: low	
Access Conditions: READ ALW UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 - 3	Emergency Call Code 1	O	3 bytes
4 - 6	Emergency Call Code 2	O	3 bytes
(3n-2) to 3n	Emergency Call Code n	O	3 bytes

- Emergency Call Code

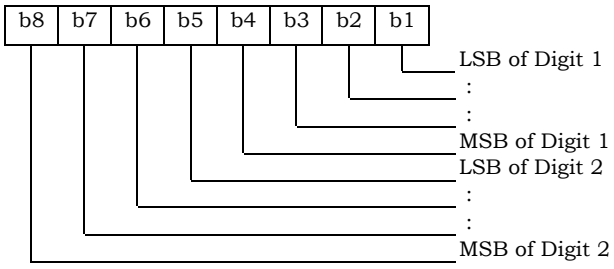
Contents:

Emergency Call Code. Each digit is encoded in BCD format.

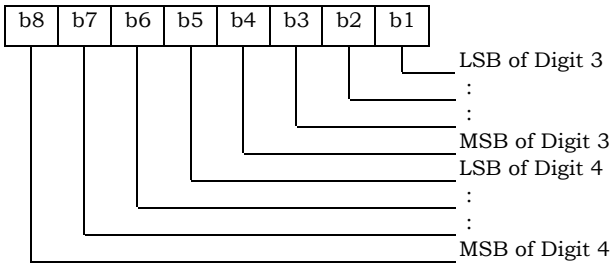
Coding:

The emergency call code is of a variable length with a maximum length of 6 digits. Each emergency call code is coded on three bytes, with each digit within the code being coded on four bits as shown below. If a code of less than 6 digits is chosen, then the unused nibbles shall be set to 'F'.

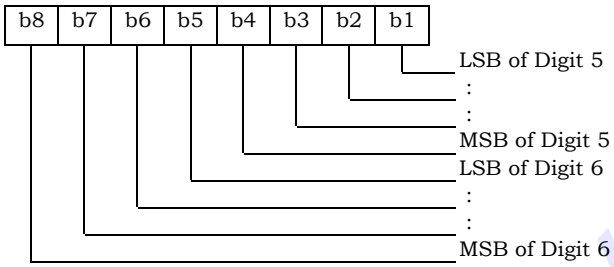
Byte 1:



Byte 2:



Byte 3:



5.2.39 $EF_{ME3GPDOPC}$ (ME 3GPD Operation Capability)

If either service n14 or n15 is available (see Section 5.2.18), this EF shall be present. This EF stores IP operation capabilities supported by the ME.

Identifier: '6F48'	Structure: transparent	Optional
File size: 1 byte	Update activity: low	

Access Conditions:

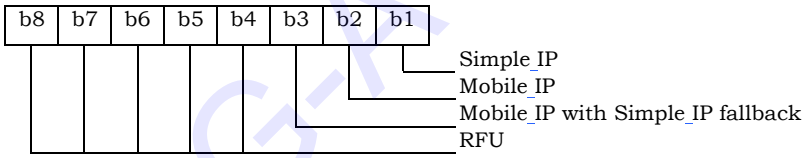
READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	see [7], 3GPD Operation Capability Parameters	M	1 byte

3GPD Operation Capability Parameters

Coding (see Section 3.5.8.1 of [7]):

Byte 1:



After the selection of ADF_{CSIM} during the initialization, the CSIM shall set the value of this byte to "0". An ME that supports Simple IP or Mobile IP shall set each subfield to '1' if it supports the corresponding operating mode.

5.2.40 $EF_{3GPDOPM}$ (3GPD Operation Mode)

If either service n14 or n15 is available (see Section 5.2.18), this EF shall be present. This EF stores the 3GPD Operation Mode Parameter Block defined in [7].

Identifier: '6F49'	Structure: transparent	Optional
File size: 1 byte	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	See [7], 3GPD Operation Mode Parameter Block	M	1 byte

Coding:

Byte 1:

5.2.41 EF_{SIPCAP} (Simple IP Capability Parameters)

If service n14 is available (see Section 5.2.18), this EF shall be present. This EF stores the SimpleIP Capability Parameter Block defined in [7].

Identifier: '6F4A'	Structure: transparent	Optional
File size: 4 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 – 4	See [7], SimpleIP Capability Parameter Block	M	4 bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.42 EF_{MIPCAP} (Mobile IP Capability Parameters)

If service n15 is available (see Section 5.2.18), this EF shall be present. This EF stores the MobileIP Capability Parameter Block defined in [7].

Identifier: '6F4B'	Structure: transparent	Optional
File size: 5 bytes	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1-5	See [7], MobileIP Capability Parameter Block	M	5 bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.43 EF_{SIPUPP} (Simple IP User Profile Parameters)

If service n14 is available (see Section 5.2.18), this EF shall be present. This EF stores the SimpleIP User Profile Parameter Block defined in [7].

Identifier: '6F4C'	Structure: transparent	Optional
File size: 1+X	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Length of SimpleIP User Profile Parameter Block	M	1 byte
2 – X+1	See [7], SimpleIP User Profile Parameter Block	M	X bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.44 EF_{MIPUPP} (Mobile IP User Profile Parameters)

If service n15 is available (see Section 5.2.18), this EF shall be present. This EF stores the MobileIP User Profile Parameter Block defined in [7].

Identifier: '6F4D'	Structure: transparent	Optional
File size: 1+X	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Length of MobileIP User Profile Parameter Block	M	1 byte
2 – X+1	See [7], MobileIP User Profile Parameter Block	M	X bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.45 EF_{SIPSP} (Simple IP Status Parameters)

If service n14 is available (see Section 5.2.18), this EF shall be present. This EF stores the SimpleIP Status Parameters Block defined in [7].

Identifier: '6F4E'	Structure: transparent	Optional
File size: 1	Update activity: low	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	See [7], SimpleIP Status Parameters Block	M	1 byte

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.46 EF_{MIPSP} (Mobile IP Status Parameters)

If service n15 is available (see Section 3.4.18), this EF shall be present. This EF stores the MobileIP Status Parameters Block defined in [7].

Identifier: '6F4F'	Structure: transparent	Optional
File size: X	Update activity: low	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 – X	See [7], MobileIP Status Parameters Block	M	X bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.47 $EF_{SIPPAPSS}$ (Simple IP PAP SS Parameters)

If service n14 is available (see Section 3.4.18), this EF shall be present. This EF stores the SimpleIP PAP SS Parameter Block defined in [7].

Identifier: '6F50'	Structure: transparent	Optional
File size: 1+X	Update activity: low	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Length of SimpleIP PAP SS Parameter Block	M	1 byte
2 – X+1	See [7], SimpleIP PAP SS Parameter Block	M	X bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.48 Reserved

5.2.49 Reserved

5.2.50 EF_{PUZL} (Preferred User Zone List)

This EF stores the Preferred User Zone List, as described in Section ~~3.5.6.4~~3.5.7 of [7].

Identifier: '6F53'	Structure: transparent	Optional
File size: '~~CUR~~MAX_UZ_LIST_SIZE'	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1- CUR_UZ_LIST_SIZE	PUZL (see Section ~~3.5.6.4~~3.5.7 of [7])	M	CUR_UZ_LIST_SIZE

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.51 $EF_{MAXPUZL}$ (Maximum PUZL)

This EF stores the maximum size, in octets, that the CSIM can support for EF_{PUZL} (See 3.5.7 of [7] for more details) and the maximum number of User Zone entries that the CSIM can support for EF_{PUZL} (See ~~3.5.6.1.~~3.5.6.1 of [7] for more details).

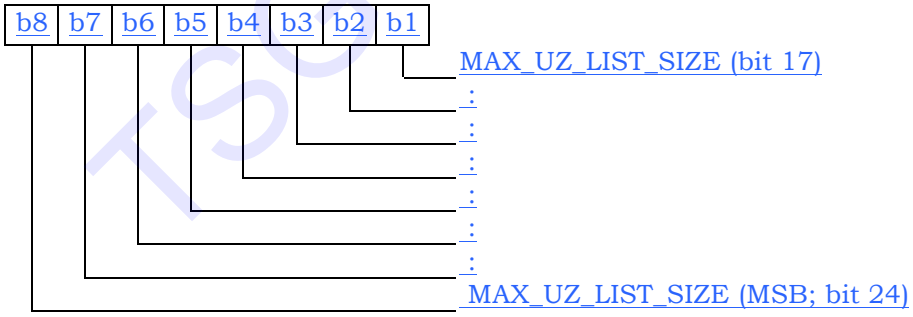
Identifier: '6F54'	Structure: transparent	Optional
File size: 5 bytes	Update activity: Never	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 –3	MAX_UZ_LIST_SIZE	M	3 bytes
4 - 5	MAX_NUM_UZ	M	2 bytes

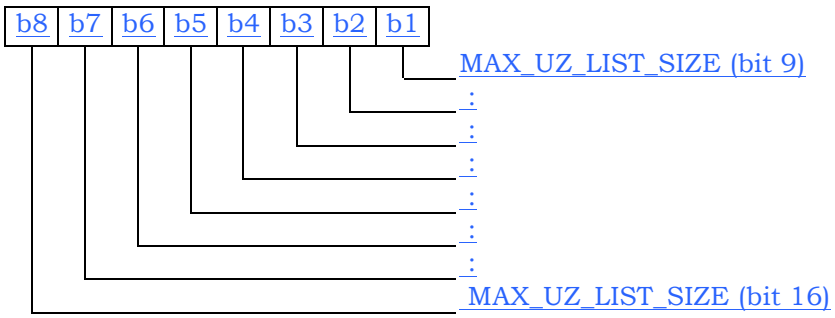
This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

Coding:

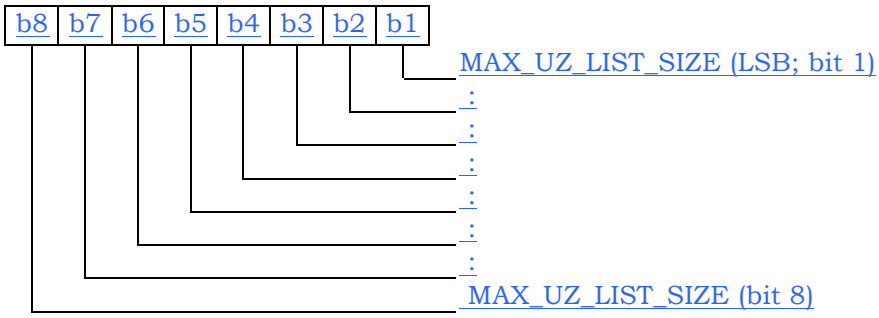
Octet 1:



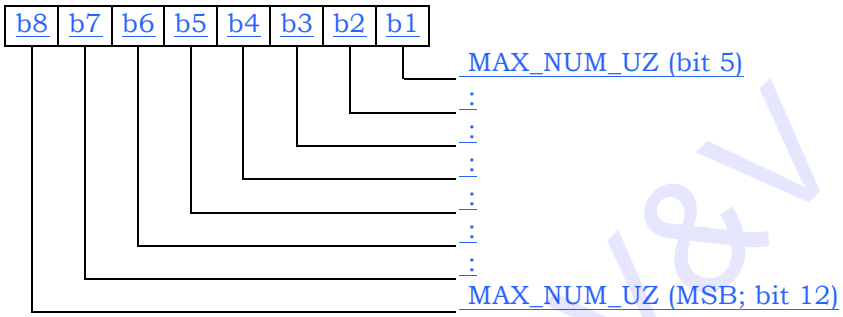
Octet 2:



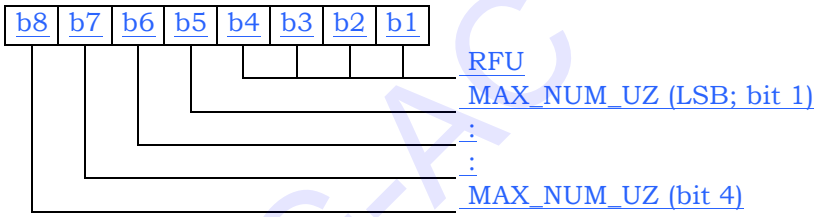
Octet 3:



Octet 4:



Octet 5:



5.2.52 EF_{MECRP} (ME-specific Configuration Request Parameters)

This EF stores ME-specific parameters to be used to form the response to the ~~Configuration Request~~CONFIGURATION REQUEST command while secure mode is active. The ME shall update these ME-specific parameters during initializations.

Identifier: '6F55'	Structure: transparent	Mandatory
File size: 3 bytes	Update activity: ~~low~~medium	

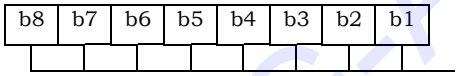
Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	SCM	M	1 byte
2	MOB_P_REV	M	1 byte
3	Local Control	M	1 byte

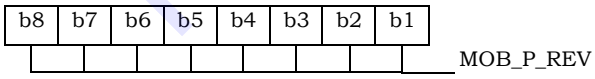
Coding:

Byte 1:

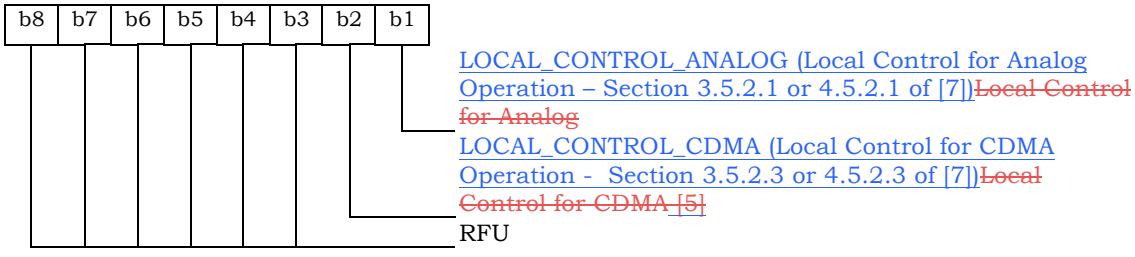


Note: b6 indicates if the ME is operating in slotted mode.

Byte 2:



Byte 3:



5.2.53 $EF_{HRPDCAP}$ (HRPD Access Authentication Capability Parameters)

If service n8 is available (see Section 5.2.18), this EF shall be present. This EF stores the HRPD Access Authentication Capability Parameters Block defined in Section 3.5.8.12 of [7].

Identifier: '6F56'	Structure: transparent	Optional
File size: 3 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 – 3	See [7], HRPD Access Authentication Capability Parameters Block	M	3 bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.54 $EF_{HRPDUPP}$ (HRPD Access Authentication User Profile Parameters)

If service n8 is available (see Section 5.2.18), this EF shall be present. This EF stores the HRPD Access Authentication User Profile Parameters Block defined in Section 3.5.8.13 of [7].

Identifier: '6F57'	Structure: transparent	Optional
File size: 1+X bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Length of HRPD Access Authentication User Profile Parameters Block	M	1 byte
2 – X+1	See [7], HRPD Access Authentication User Profile Parameters Block	M	X bytes

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.55 EF_{CSSPR} (CUR_SSPR_P_REV)

This EF stores the protocol revision (CUR_SSPR_P_REV) of the current extended preferred roaming list stored in the EF_{EPRL}. This information, described in Section 3.5.3.3 of [7], is used by the ME to parse the EF_{EPRL}.

Identifier: '6F58'	Structure: transparent	Optional
File size: 1	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	CUR_SSPR_P_REV	M	1 byte

Notes:

1. It is recommended that CUR_SSPR_P_REV in Octet 7 of EF_{EPRL} (as defined in section 3.5.3.3 of [7]) be used instead of this EF_{CSSPR}.
2. According to [7], CUR_SSPR_P_REV is used to indicate if the PRL or EPRL is stored in PR_LISTs-p and according to section 3.3.1.3 of [7], the MS shall store CUR_SSPR_P_REV for not only the PRL but also the EPRL after an SSPR Download Request. However, since a CSIM can store the PRL and EPRL in EF_{PRL} and EF_{EPRL}, respectively, there is no need to distinguish what is stored in EF_{PRL}. Hence, EF_{CSSPR} is only applicable for EF_{EPRL} and not EF_{PRL} as [7] would seem to require.

5.2.56 EF_{ATC} (Access Terminal Class)

If service n8 is available (see Section 5.2.18), this EF shall be present. This EF stores the class of access terminal used for Persistence Test in the system defined in [28].

Identifier: '6F59'	Structure: transparent	Optional
File size: 1	Update activity: low	

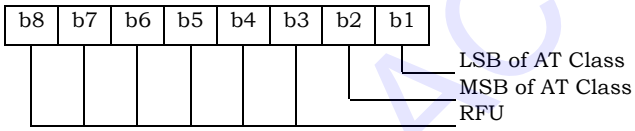
Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Access Terminal Class	M	1 byte

Coding:

Byte 1:



5.2.57 EF_{EPRL} (Extended Preferred Roaming List)

This EF stores the Extended Preferred Roaming List, as described in Section ~~3.5.3~~3.5.5 of [7].

Identifier: '6F5A'	Structure: transparent	Optional
SFI: '0E'		
File size: 'MAX_PR_LIST_SIZE' for EF_{EPRL}	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-PR_LIST_SIZE	PR_LIST (see Section 3.5.5 of [7])	M	PR_LIST_SIZE

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

- PR_LIST

  Contents:

  The Extended Preferred Roaming List.

  Coding:

  As defined in section 3.5.5 of [7].

5.2.58 $EF_{BCSMScfg}$ (Broadcast Short Message Configuration)

If service n9 is available, this EF shall be present.

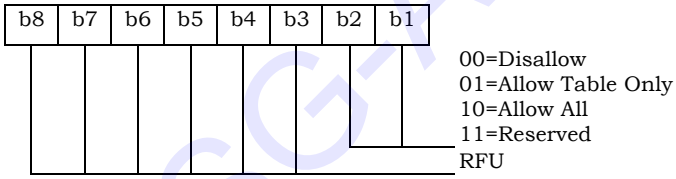
This EF contains the operator broadcast configuration setting for Broadcast SMS. This information, determined by the operator, defines the filtering criteria that can be used by the ME to receive Broadcast SMS.

Identifier: '6F5B'	Structure: transparent	Optional
File size: 1 byte	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	Operator Broadcast Configuration	M	1 byte

Coding:

Byte 1:



Operator configuration includes filtering criteria imposed by a service provider.

Field Name	Description
Disallow	This setting disables the mobile station's broadcast SMS capability (i.e., the mobile station will not process broadcast SMS).
Allow Table Only	This setting allows the mobile station to receive only broadcast messages for the service categories that have been programmed in $EF_{BCSMStable}$
Allow All	This setting allows the mobile station to receive broadcast messages for all service categories.

5.2.59 $EF_{BCSMSpref}$ (Broadcast Short Message Preference)

If service n9 is available, this EF shall be present.

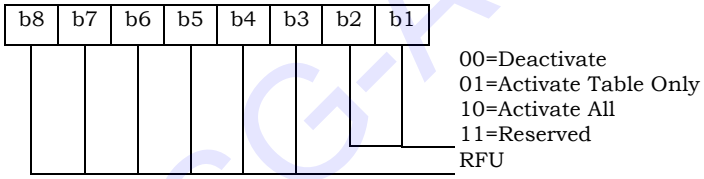
This EF contains the user broadcast configuration setting for Broadcast SMS. This information, determined by the user, defines the filtering criteria that can be used by the Mobile Equipment (ME) to receive Broadcast SMS.

Identifier: '6F5C'	Structure: transparent	Optional
File size: 1 byte	Update activity: high	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1	User Broadcast Configuration	M	1 byte

Coding:

Byte 1:



User configuration includes filtering criteria determined by the mobile user.

Field Name	Description
Deactivate	This setting deactivates the mobile station's broadcast SMS functions (i.e., the mobile station will not process broadcast SMS).
Activate Table Only	This setting allows the mobile station to receive only broadcast messages for the service categories that have been programmed in EFBCSMStable, subject to any additional filtering criteria included in EFBCSMStable based on user preferences. This setting is only valid if the operator configuration is not Disallow. Moreover, the mobile user can selectively enable and disable individual programmed entries in EFBCSMStable.
Activate All	This setting allows the mobile station to receive broadcast messages for all service categories. This setting is only valid if the operator configuration is "Allow All". EFBCSMStable will not be consulted for this setting.

5.2.60 $EF_{BCSMStable}$ (Broadcast Short Message Table)

If service n9 is available, this EF shall be present.

This EF contains information in accordance with [8] comprising service category program parameters, which can be used by the Mobile Equipment (ME) for Broadcast SMS filtering. See Section 4.5.19 of [8] for more detail.

Each record in this EF is linked to a record with the same record index in EF_{BCSMSP}.

Identifier: '6F5D'	Structure: linear fixed	Optional
Record Length: 7+X byte	Update activity: high	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Status	M	1 byte
2 – 3	Service Category	M	2 bytes
4	Language	M	1 byte
5	Max Messages	M	1 byte
6	Alert Option	M	1 byte
7	Label Encoding	M	1 byte
8 to 7+X	Label	M	X byte

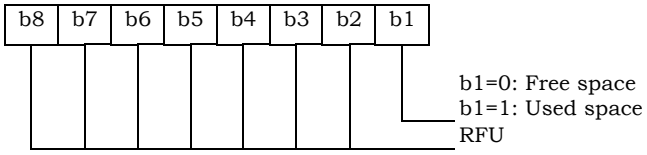
- Status

Contents:

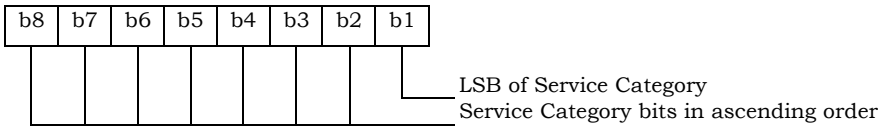
Status byte of the record which can be used as a pattern in the SEEK command.

Coding:

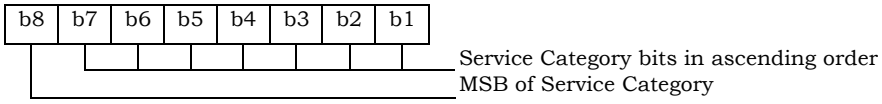
Byte 1:



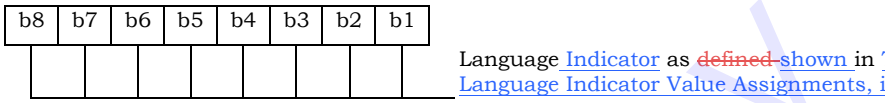
Byte 2:



Byte 3:

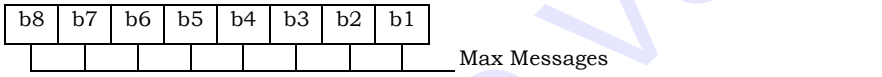


Byte 4:

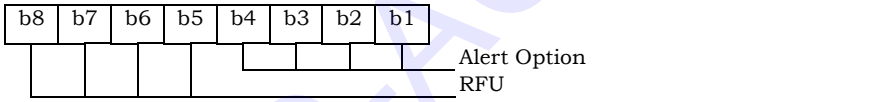


Language Indicator as ~~defined~~ shown in Table 9.2.1, Language Indicator Value Assignments, in [Informative 1].

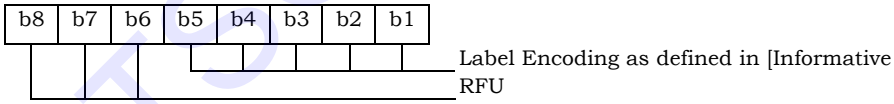
Byte 5:



Byte 6:



Byte 7:



5.2.61 EF_{BCSMSP} (Broadcast Short Message Parameter)

If service n9 is available, this EF shall be present.

This EF contains selection flag and priority associated with service categories and used by the ME for filtering of BC-SMS. Each record in this EF is linked to a record with the same record index in $EF_{BCSMStable}$.

Identifier: '6F5E'	Structure: linear fixed	Optional
Record Length: 2 bytes	Update activity: high	

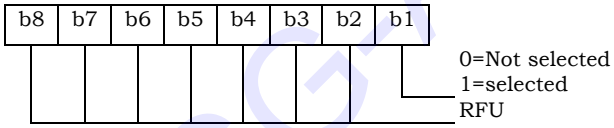
Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

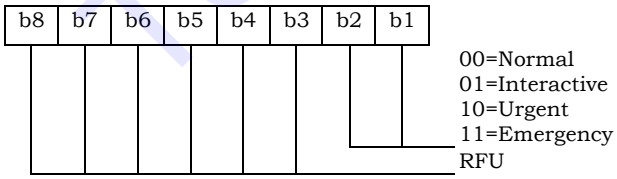
Bytes	Description	M/O	Length
1	Select	M	1 byte
2	Priority	M	1 byte

Coding:

Byte 1:



Byte 2:



Unused records are filled with 'FF'. When the b1 of Byte 1 is set to '1', then the ME shall filter the BC-SMS according to the priority indicated in Byte 2.

5.2.62 $EF_{BAKPARA}$ (Currently used BAK Parameters)

If service n18 is available, this EF shall be present.

This EF contains BCMCS related parameters, i.e.: BCMCS_Flow_ ID, BAK_ ID and BAK_Expire, corresponding to BAK keys that have been delivered to the CSIM and are currently used. See [36] for more details.

Identifier: '6F63'	Structure: Linear Fixed	Optional
Record length: X+Y+Z+3 bytes	Update activity: high	

Access Conditions:

READ	PIN
UPDATE	ADM
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1	Length of BCMCS_Flow_ID	M	1 byte
2 to X +1	BCMCS_Flow_ID	M	X bytes
X+2	Length of BAK_ID	M	1 byte
X+3 to X+Y+2	BAK_ID	M	Y bytes
X+Y+3	Length of BAK_Expire	M	1 byte
X+Y+4 to X+Y+Z+3	BAK_Expire	M	Z bytes

- Length of BCMCS_Flow_ID

  Content: number of bytes of the following data item containing the BCMCS flow identifier.
  Coding: Binary.

- BCMCS_Flow_ID

  Content: BCMCS Flow Identifier

  Coding: Binary.

- Length of BAK_ID

  Content: number of bytes of the following data item containing the BAK identifier.
  Coding: Binary

- BAK_ID

  Content: BAK Identifier

Coding: Binary.

- Length of BAK_Expire

Content: number of bytes of the following data item containing the BAK_Expire.

Coding: Binary

- BAK_Expire

Content: BAK_Expire

Coding: Binary.

5.2.63 $EF_{UPBAKPARA}$ (Updated BAK Parameters)

If service n18 is available, this EF shall be present.

This EF contains BCMCS related parameters, i.e.: BCMCS_Flow_ID, BAK_ID and BAK_Expire, corresponding to BAK keys that have been delivered to the CSIM but have not yet been used. See [36] for more details.

Identifier: '6F64'	Structure: cyclic	Optional
Record length: X+Y+Z+3 bytes	Update activity: high	

Access Conditions:

READ	PIN
UPDATE	ADM
DEACTIVATE	ADM
ACTIVATE	ADM

Bytes	Description	M/O	Length
1	Length of BCMCS_Flow_ID	M	1 byte
2 to X +1	BCMCS_Flow_ID	M	X bytes
X+2	Length of BAK_ID	M	1 byte
X+3 to X+2+Y	BAK_ID	M	Y bytes
X+Y+3	Length of BAK_Expire	M	1 byte
X+Y+4 to X+Y+Z+3	BAK_Expire	M	Z bytes

- Length of BCMCS_Flow_ID

  Content: number of bytes of the following data item containing the BCMCS flow identifier.
  Coding: Binary

- BCMCS_Flow_ID

  Content: BCMCS Flow Identifier

  Coding: Binary.

- Length of BAK_ID

  Content: number of bytes of the following data item containing the BAK identifier.
  Coding: Binary

- BAK_ID

  Content: BAK Identifier

Coding: Binary.

- Length of BAK_Expire

Content: number of bytes of the following data item containing the BAK_Expire.

Coding: Binary

- BAK_Expire

Content: BAK_Expire

Coding: Binary.

5.2.64 EF_{MMSN} (MMS Notification)

If service n19 is available, this file shall be present.

This EF contains information in accordance with [37] comprising MMS notifications (and associated parameters) which have been received by the ME from the network.

Identifier: '6F65'	Structure: Linear fixed	Optional
Record length: ~~4+~~X bytes	Update activity: low	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 - 2	MMS Status	M	2 bytes
3	MMS Implementation	M	1 byte
4 to ~~X~~Y+3	MMS Notification	M	~~X~~ Y bytes
~~X~~Y+4	Extension file record number	M	1 byte

Note: X ≥ Y+4 for every record.

- MMS Status

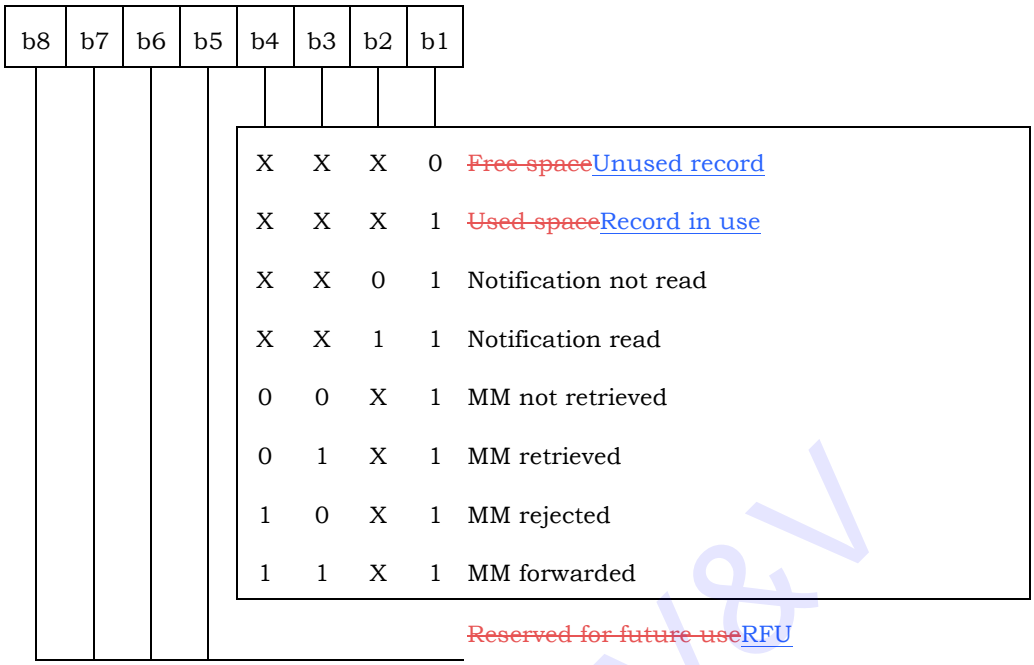
Content:

The status bytes contain the status information ~~of~~ for the notification.

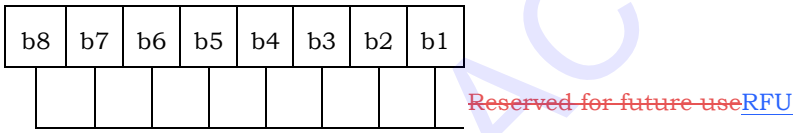
~~Coding:~~

- ~~b1 indicates whether there is valid data or if the location is free.~~
- ~~b2 indicates whether the MMS notification has been read or not.~~
- ~~b3 and b4 of the first byte indicate the MM retrieval, MM rejection, or MM forwarding status.~~
- ~~b5 to b8 of the first byte and the entire second byte are reserved for future use.~~

~~First byte~~Octet 1:



~~Second byte~~Octet 2:

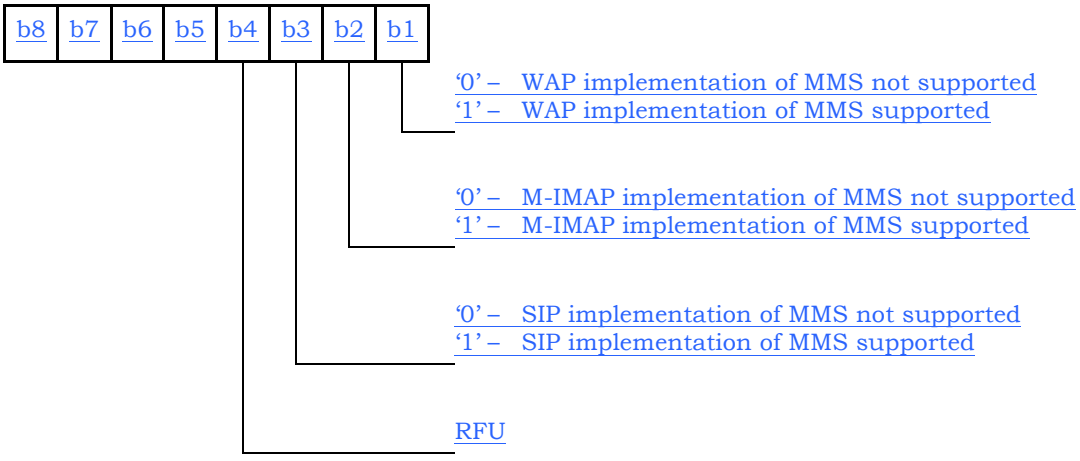


- MMS Implementation

Contents:

The MMS Implementation indicates the used implementation type, e.g. WAP, M-IMAP, SIP.

Octet 3:



~~Coding:~~

~~Allocation of bits:~~

~~• Bit number Parameter indicated~~
~~1 WAP implementation of MMS~~
~~2 M-IMAP implementation of MMS~~
~~3 SIP implementation of MMS~~
~~4-8 Reserved for future use~~

~~• Bit value Meaning~~
~~0 Implementation not supported.~~
~~1 Implementation supported.~~

- MMS Notification

  Contents:

  The MMS Notification contains the MMS notification.

  Coding:

  The MMS Notification is coded according to the MMS Implementation as indicated in ~~Byte~~ octet 3.

  Any unused ~~byte~~ octets shall be set to 'FF'.

- Extension file record number

  Contents:

  - extension file record number. This ~~byte~~ octet identifies the number of a record in the EF_{EXT8} containing extension data for the notification information. The use of this ~~byte~~ octet is optional. If it is not used it shall be set to 'FF'.

  Coding:

  - binary.

5.2.65 EF_{EXT8} (Extension 8)

If service n20 is available, this file shall be present.

This EF contains extension data of a MMS Notification (Multimedia Messaging Service).

Identifier: '6F66'	Structure: linear fixed	Optional
Record length: X+2 bytes	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Record type	M	1 byte
2 to X+1	Extension data	M	X bytes
X+2	Identifier	M	1 byte

For contents and coding see [30].

5.2.66 EF_{MMSICP} (MMS Issuer Connectivity Parameters)

If service n19 is available, this file shall be present.

This EF contains values for Multimedia Messaging Connectivity Parameters as determined by the issuer, which can be used by the ME for MMS network connection. This file may contain one or more sets of Multimedia Messaging Issuer Connectivity Parameters. The first set of Multimedia Messaging Issuer Connectivity Parameters is used as the default set.

Each set of Multimedia Messaging Issuer Connectivity Parameters may consist of one or more "Interface to Core Network and Bearer information" TLV objects (only for WAP), but shall contain only one "MMS Implementation" TLV object (for WAP, M-IMAP and SIP), one "MMS Relay/Server" TLV object (for WAP, M-IMAP and SIP) and one "Gateway" TLV object (only for WAP).

The order of the "Interface to Core Network and Bearer information" TLV objects in the MMS Connectivity TLV object defines the priority of the Interface to Core Network and Bearer information, with the first TLV object having the highest priority.

Identifier: '6F67'	Structure: Transparent	Optional
File Size: X_1+…+ X_n bytes	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to X_1	MMS Connectivity Parameters TLV object	M	X_1 bytes
X_1+1 to X_1 + X_2	MMS Connectivity Parameters TLV object	O	X_2 bytes
…	…		
X_1+…+ X_{n-1}+1 to X_1+…+ X_n	MMS Connectivity Parameters TLV object	O	X_n bytes

- MMS Connectivity Parameters tags

Description	Tag Value
MMS Connectivity Parameters Tag	'AB'
MMS Implementation Tag	‘80’
MMS Relay/Server Tag	‘81’
Interface to Core Network and Bearer Information Tag	'82'
Gateway Tag	'83'
MMS Authentication Mechanism Tag	‘84’
MMS Authentication ID Tag	‘85’

- MMS Connectivity Parameters contents

Description	Value	M/O	Length (bytes)
MMS Connectivity Parameters Tag	'AB'	M	1
Length	Note 1	M	Note 2
MMS Implementation Tag	'80'	M	1
Length	1	M	1
MMS Implementation Information	--	M	1
MMS Relay/Server Tag	'81'	M	1
Length	X	M	Note 2
MMS Relay/Server Address	--	M	X
~~1st~~ First Interface to Core Network and Bearer Information Tag (highest priority)	'82'	C2	1
Length	Y1	C2	Note 2
First ~~1st~~ Interface to Core Network and Bearer information	--	C2	Y1
~~2nd~~ Second Interface to Core Network and Bearer Information Tag	'82'	C2	1
Length	Y2	C2	Note 2
Second ~~2nd~~ Interface to Core Network and Bearer information	--	C2	Y2
…			
Nth Interface to Core Network and Bearer Information Tag (lowest priority)	'82'	C2	1
Length	Y3	C2	Note 2
Nth Interface to Core Network and Bearer information	--	C2	Y3
Gateway Tag	'83'	O	1
Length	Z	O	Note 2
Gateway Information	--	O	Z
MMS Authentication Mechanism Tag	'84'	C1	1
Length	X	C1	Note 2
MMS Authentication Mechanism	--	C1	X
MMS Authentication ID Tag	'85'	C1	1
Length	X	C1	Note 2
MMS Authentication ID (Login_ID)	--	C1	X
NOTE 1: This is the total size of the constructed TLV object (not including the tag and this length).			
NOTE 2: The length is coded according to [60] using primitive encoding and the minimum number of octets~~ISO/IEC 8825~~.			
C1: only present if M-IMAP or SIP indicated in tag 80			
C2: only present if WAP is indicated in tag 80			

- MMS Implementation Tag '80'

  See [30] for contents and coding.

- MMS Relay/server Tag '81'

  Contents:

  The MMS relay/server contains the address of the associated MMS relay/server; In addition, for M-IMAP and SIP, authentication mechanism and authentication ID (Login ID) are also included.

  Coding:

  The MMS relay/server address is coded as URI appropriate to the MM1 implementation being used, for example SIP, or M-IMAP.

- Interface to Core Network and Bearer Information Tag '82'

  Contents:

  The Interface to Core Network and Bearer Information may contain the following information to set up the bearer: Bearer, Address, Type of address, Speed, Call type, Authentication type, Authentication id, Authentication password.

  Coding:

  The coding is according to the guideline provided in [37]. If MMS implementation type is WAP, all instances of Interface to Core Network and Bearer Information are optional. If MMS implementation type is M-IMAP or SIP, no Interface to Core Network and Bearer Information is needed.

- Gateway Tag '83'

  Contents:

  The Gateway may contain the following information; Address, Type of address, Port, Service, Authentication type, Authentication id and Authentication password.

  Coding:

  The coding is according to the guideline provided in [37].

- MMS Authentication Mechanism Tag ‘84’

  Contents:

  The MMS authentication mechanism contains the authentication mechanism for MMS. It is mandatory for M-IMAP and SIP.

  Coding:

  The MMS authentication mechanism is coded as in Section 4.10.1 of [46].

- MMS Authentication ID Tag ‘85’

  Contents:

  The MMS authentication ID contains the authentication ID for MMS. It is mandatory for M-IMAP and SIP.

  Coding:

  The coding is according to the guideline provided in [37].

Unused bytes shall be set to 'FF'.

5.2.67 EF_{MMSUP} (MMS User Preferences)

If service n19 is available, this file shall be present.

This EF contains values for Multimedia Messaging Service User Preferences, which can be used by the ME for user assistance in preparation of mobile multimedia messages (e.g. default values for parameters that are often used).

Identifier: '6F68'	Structure: Linear Fixed	Optional
Record Length: X bytes	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	PIN
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to X	MMS User Preference TLV Objects	M	X bytes

- MMS User Preference tags

Description	Tag Value
MMS Implementation Tag	'80'
MMS User preference profile name Tag	'81'
MMS User Preference information Tag	'82'

- MMS User Preference information

Description	Value	M/O	Length (bytes)
MMS Implementation Tag	'80'	M	1
Length	1	M	~~Note~~1
MMS Implementation information	--	M	1
MMS User preference profile name Tag	'81'	M	1
Length	~~X~~Y	M	Note
MMS User profile name	--	M	~~X~~Y
MMS User Preference information Tag	'82'	M	1
Length	~~Y~~Z	M	Note
MMS User Preference information	--	M	~~Y~~Z
NOTE: The length is coded according to [60] using primitive encoding and the minimum number of octets~~ISO/IEC 8825~~.			

- MMS Implementation Tag '80'

  For contents and coding see [30].

- MMS User preference profile name Tag '81'

  Contents:

  Alpha tagging of the MMS user preference profile.

  Coding:

  This alpha-tagging shall use either:
  - the SMS default 7-bit coded alphabet as defined in [38] with bit 8 set to 0. The alpha identifier shall be left justified; or
  - one of the UCS2 coded options as defined in the annex of [30].

- MMS User Preference information Tag '82'

  Contents:

  The following information elements may be coded; Sender Visibility, Delivery Report, Read-Reply, Priority, Time of Expiry and Earliest Delivery Time. Refer to [37], [39], [40], and [41].

  Coding:

  Depending upon the MMS implementation as indicated in Tag '80'.

5.2.68 EF_{MMSUCP} (MMS User Connectivity Parameters)

If service n19 and n21 are available, this file shall be present.

This EF contains values for Multimedia Messaging Connectivity Parameters as determined by the user, which can be used by the ME for MMS network connection. This file may contain one or more sets of Multimedia Messaging User Connectivity Parameters.

Each set of Multimedia Messaging User Connectivity Parameters may consist of one or more "Interface to Core Network and Bearer information" TLV objects (only for WAP), but shall contain only one "MMS Implementation" TLV object (for WAP, M-IMAP and SIP), one "MMS Relay/Server" TLV object (for WAP, M-IMAP and SIP) and one "Gateway" TLV object (only for WAP).

The order of the "Interface to Core Network and Bearer information" TLV objects in the MMS Connectivity TLV object defines the priority of the Interface to Core Network and Bearer information, with the first TLV object having the highest priority.

Identifier: '6F69'	Structure: Transparent	Optional
File Size: X_1+…+ X_n bytes	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	PIN/PIN2 (fixed during administrative management)
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to X_1	MMS Connectivity Parameters TLV object	O	X_1 bytes
X_1+1 to X_1 + X_2	MMS Connectivity Parameters TLV object	O	X_2 bytes
…	…		
X_1+…+ X_{n-1}+1 to X_1+…+ X_n	MMS Connectivity Parameters TLV object	O	X_n bytes

For the contents and coding see Section 5.2.65 EF_{MMSICP}.

5.2.69 $EF_{AuthCapability}$ (Authentication Capability)

If service n22 is available, this file shall be present. This EF stores authentication capabilities for each application supported by the CSIM.

Identifier: '6F6A'	Structure: Linear Fixed	Optional
Record Length: 5 bytes	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1	Application ID	M	1 byte
2-3	Authentication Capability	M	2 bytes
4-5	~~Reserved~~RFU	M	2 bytes

Coding:

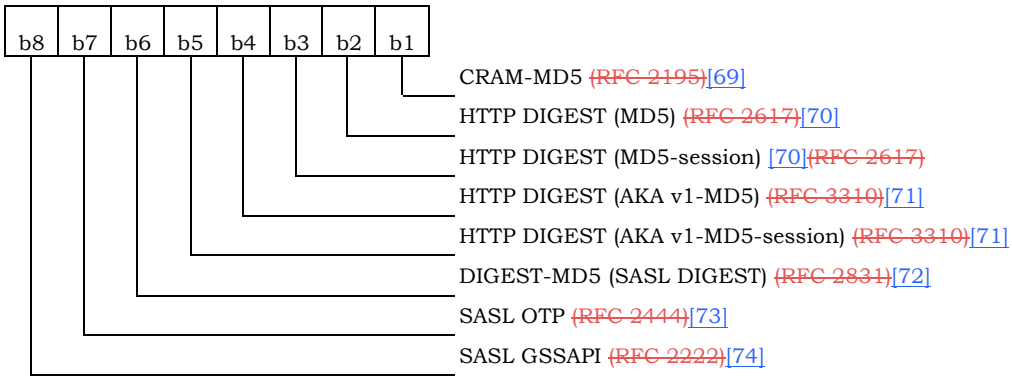
Byte 1:

The coding for Application ID is as follows:[6]

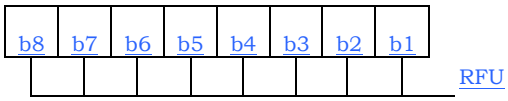
Binary Value	**Application ID**
'00000000'	MMS
'0000001'-'11111111'	Reserved

[6] Note that the Application ID for MMD is not listed for CSIM in contrast to [46] where it is listed – as MMD functions are defined in ISIM.

Byte 2:



Byte~~s~~ 3~~-5 are reserved.~~:



The CSIM shall set each subfield to '1' if it supports the corresponding authentication mechanism.

5.2.70 EF_{3GCIK} (3G Cipher and Integrity Keys)

If service n16 is available, this file shall be present.

This EF contains the cipher key (CK) and, the integrity key (IK) produced by the '3G Access AKA' AUTHENTICATE command.

Identifier : '6F6B'	Structure : transparent	Optional
SFI: '0B'		
File size: 32 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 - 16	Cipher key (CK)	M	16 bytes
17 - 32	Integrity key (IK)	M	16 bytes

- Cipher key (CK).

  Coding:

  The least significant bit of CK is the least significant bit of the 16th byte. The most significant bit of CK is the most significant bit of the 1st byte.

- Integrity key (IK).

  Coding:

  The least significant bit of IK is the least significant bit of the 32nd byte. The most significant bit of IK is the most significant bit of the 17th byte.

5.2.71 EF_{DCK} (De-Personalization Control Keys)

If service n25 is available, this EF shall be present.

This EF provides storage for the de-personalization control keys associated with the OTA de-personalization cycle of [44].

Identifier: '6F6C'	Structure: transparent	Optional
File size: 20 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to 4	8 digits of Network Type 1 de-personalization control key	M	4 bytes
5 to 8	8 digits of Network Type 2 de-personalization control key	M	4 bytes
9 to 12	8 digits of service provider de-personalization control key	M	4 bytes
13 to16	8 digits of corporate de-personalization control key	M	4 bytes
17 to 20	8 digits of HRPD Network de-personalization control key	M	4 bytes

Empty control key fields shall be coded 'FFFFFFFF'.

5.2.72 EF_{GID1} (Group Identifier Level 1)

If service n23 is available, this EF shall be present.

This EF contains identifiers for particular CSIM/ME associations. It can be used to identify a group of CSIMs for a particular application.

Identifier: '6F6D'	Structure: transparent	Optional
File size: 1 to n bytes	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to n	CSIM group identifier(s)	O	n bytes

5.2.73 EF_{GID2} (Group Identifier Level 2)

If service n24 is available, this EF shall be present.

This EF contains identifiers for particular CSIM/ME associations. It can be used to identify a group of CSIMs for a particular application.

Identifier: '6F6E'	Structure: transparent	Optional
File size: 1 to n bytes	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to n	CSIM group identifier(s)	O	n bytes

NOTE: The structure of EF_{GID1} and EF_{GID2} are identical. They are provided to allow the network operator to enforce different levels of security dependant on an application.

5.2.74 $EF_{CDMACNL}$ (CDMA Co-operative Network List)

If service n26 is available, this EF shall be present.

This EF contains the Co-operative Network List for the multiple network personalization services defined in [44].

Identifier: '6F6F'	Structure: transparent	Optional
File size: 7n bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to 7	Element 1 of co-operative net list	M	7 bytes
...			
7n-6 to 7n	Element n of co-operative net list	O	7 bytes

- Co-operative Network List

  Contents:

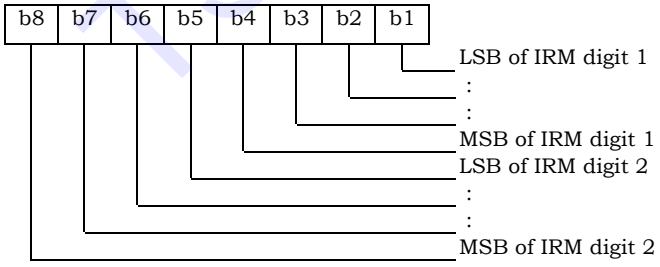
  Service provider ID and corporate ID of co-operative networks.

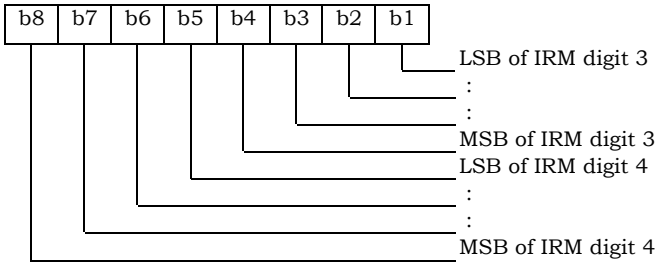
  Coding:

  For each 7 byte list element:

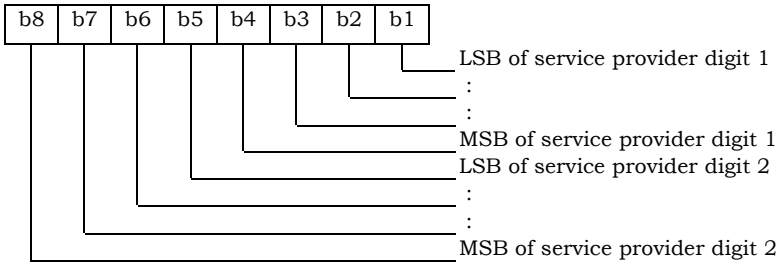
  Byte 1 to 3: MCC + MNC: As per ~~ITU-T Recommendation E.212~~ Annex A of [9].

  Byte 4 to 5: 4 most significant digits of the International Roaming based MIN.

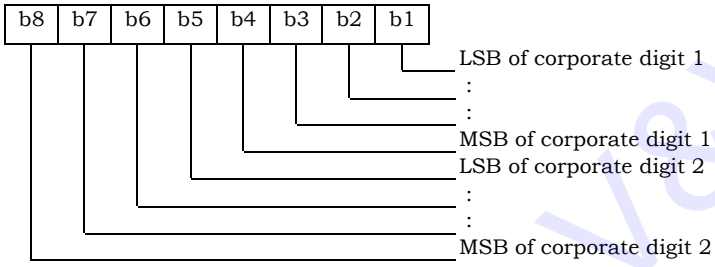




Byte 6:



Byte 7:



Empty fields shall be coded with 'FF'.

The end of the list is delimited by the first MCC field coded 'FFF'.

5.2.75 EF_{HOME_TAG} (Home System Tag)

This EF stores the Home System Tag, as described in Section 3.5.10.1 of [7].

Identifier: '6F70'	Structure: transparent	Mandatory
File size: X bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 - X	Home System Tag (see Section 3.5.10.1 of [7])	M	Variable

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.76 EF_{GROUP_TAG} (Group Tag List)

This EF stores the Group Tag List, as described in Section ~~3.5.11~~3.5.10.3 of [7].

Identifier: '6F71'	Structure: transparent	Mandatory
File size: GROUP_TAG_LIST_SIZE	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-GROUP_TAG_LIST_SIZE	Group Tag List (see Section ~~3.5.11~~3.5.10.3 of [7])	M	Variable

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.77 EF$_{SPECIFIC_TAG}$ (Specific Tag List)

This EF stores the Specific Tag List, as described in Section ~~3.5.11~~3.5.10.5 of [7].

Identifier: '6F72'	Structure: transparent	Mandatory
File size: SPEC_TAG_LIST_SIZE'	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-SPEC_TAG_LIST_SIZE	Specific Tag List (see Section ~~3.5.11~~3.5.10.5 of [7])	M	Variable

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.78 EF_{CALL_PROMPT} (Call Prompt List)

This EF stores the Call Prompt List, as described in Section ~~3.5.11~~3.5.10.7 of [7].

Identifier: '6F73'	Structure: transparent	Mandatory
File size: 'CALL_PRMPT_LIST_SIZE'	Update activity: low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1-CALL_PRMPT_LIST_SIZE	Call Prompt List (see Section ~~3.5.11~~3.5.10.7 of [7])	M	Variable

This EF is stored using the convention from [7], i.e. fields are placed into octets starting with the MSB of the first field into bit 8 of the first octet, followed by the remaining fields placed in sequence into the remaining bits allocated for those fields. A multi-octet integer is stored by placing the octet with the MSB into the lowest numbered available octet allocated for that integer in the EF.

5.2.79 EF_{SF_EUIMID} (Short Form EUIMID)

If service n34 is available, this file shall be present.

This EF stores the 56-bit electronic identification number (ID) unique to the CSIM.

The order of the digits when treated as 14 four-bit digits is shown in the table below, with 'd1' representing the leftmost/most significant digit and 'd14' representing the rightmost/least significant digit.

Identifier: '6F74'	Structure: transparent	Optional
File size: 7 bytes	Update activity: low	

Access Conditions:

READ	ALW
UPDATE	Never
INVALIDATE	Never
REHABILITATE	Never

	Description									
Bytes	8	7	6	5	4	3	2	1	M/O	Length
1	d13				d14				M	1 byte
2	d11				d12				M	1 byte
3	d9				d10				M	1 byte
4	d7				d8				M	1 byte
5	d5				d6				M	1 byte
6	d3				d4				M	1 byte
7	d1				d2				M	1 byte

5.2.80 EF_{EST} (Enabled Service Table)

If service n2 is “available” (as indicated in the CSIM Service Table), this file shall be present.

This EF indicates which services are enabled. If a service is not indicated as enabled in this table, the ME shall not select the service.

Identifier: '6F75'	Structure: transparent	Optional
SFI: '0F'		
File size: X bytes	Update activity: low	
Access Conditions: READ PIN UPDATE PIN2 DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1	Enabled Services n°1 to n°8	M	1 byte
2	Enabled Services n°9 to n°16	O	1 byte
etc.			
X	Enabled Services n °(8X-7) to n°(8X)	O	1 byte

Enabled Service number	Name	Corresponding CSIM Service number
1	Fixed Dialing Numbers (FDN)	2
2 – 8	Reserved	Not applicable

~~Services~~		~~CSIM Service n°~~	~~Enabled Service n°~~	
~~Contents:~~		~~2~~	~~Service n°1:~~	~~Fixed Dialling Numbers (FDN)~~

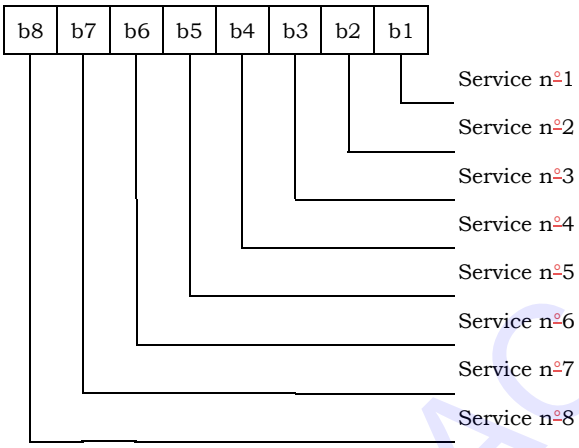
The EF shall contain at least one byte. Further bytes may be included, but if the EF includes an optional byte, then the EF shall also contain all bytes before that byte. Other services are possible in the future. The coding falls under the responsibility of the 3GPP2.

Coding:

- 1 bit is used to code each service:
  bit = 1: service ~~activated~~enabled;
  bit = 0: service ~~deactivated~~disabled.
- Unused bits shall be set to '0'.

~~A service which is listed in this table is enabled if it is indicated as available in the CSIM Service Table (CSIM_ST) and indicated as activated in the Enabled Services Tables (EST) otherwise this service is, either not available or disabled.~~

First byte:



etc.

5.2.81 $EF_{HiddenKey}$ (Key for hidden phone book entries)

This EF contains the hidden key that has to be verified by the ME in order to display the phone book entries that are marked as hidden. The hidden key can consist of 4 to 8 digits.

Identifier: '6F76'	Structure: transparent	Optional
File size: 4 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE PIN DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1 to 4	Hidden Key	M	4 bytes

- Hidden Key.

Coding:

- The hidden key is coded on 4 bytes using BCD coding. The minimum number of digits is 4. Unused digits are padded with 'F'.

- NOTE 1: Digits are not swapped, i.e. for instance the key "1234" is coded as '12 34 FF FF'.

- NOTE 2: The phone book entries marked as hidden are not scrambled by means of the hidden key. They are stored in plain text in the phone book.

5.2.82 EF_{LCSVER} (LCS Protocol Version)

If service n17 is available, this file shall be present.

This EF contains 'n' LCS Protocol Version Parameters (as defined in [50]) to indicate the version(s) of the supported protocol(s) supported by CSIM.

Each element of Protocol Version Parameter consists of 'S-SAFE Protocol version', 'TLS Session-A Protocol version', and 'TLS Session-B Protocol version'.

CSIM may support more than one version for each protocol.

Identifier: '6F77'	Structure: transparent	Optional
File size: 4n bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1 to 4	1st element of Protocol Version Parameter	M	4 bytes
...	...		..
4n-3 to 4n	nth element of Protocol Version Parameter	O	4 bytes

- Protocol Version Parameter

  Contents:

  S-SAFE Protocol version, TLS Session-A Protocol version, and TLS Session-B Protocol version.

  Coding:

  For each 4 bytes list element:

  Byte 1: S-SAFE Protocol version (LCS_S_SAFE_VERSION).
  Byte 2 to 3: TLS Session-A Protocol version (TLS client_version/server_version).
  Byte 4: TLS Session-B Protocol version (LCS_UIM_PDE_TLS_PSK_VERSION).

  Empty fields shall be coded with 'FF'.

5.2.83 EF_{LCSCP} (LCS Connectivity Parameter)

If service n17 is available, this file shall be present.

This EF contains values for IP-based LCS Connectivity Parameters as determined by the issuer, which can be used by the ME for LCS network connection.

Identifier: '6F78'	Structure: Transparent	Optional
File Size: X bytes	Update activity: low	

Access Conditions:

READ	PIN
UPDATE	ADM
INVALIDATE	ADM
REHABILITATE	ADM

Bytes	Description	M/O	Length
1 to X	LCS TLS Connectivity Parameters TLV objects	M	X bytes

LCS TLS Connectivity Parameters tags

Description	Tag Value
H-PS address (IPv4) Tag	‘80’
H-PS address (IPv6) Tag	‘81’
H-PS address (URL) Tag	‘82’

- LCS Connectivity Parameters contents

Description	Value	M/O	Length (bytes)
H-PS Address (IPv4) Tag	'80'	O	1
Length	6	O	1
H-PS IPv4 Address	--	O	4
H-PS IPv4 Port Number	--	O	2
H-PS Address (IPv6) Tag	'81'	O	1
Length	18	O	1
H-PS IPv6 Address	--	O	16
H-PS IPv6 Port Number	--	O	2
H-PS Address (URL) Tag	'82'	M	1
Length	X	M	1
H-PS URL Address	--	M	X

5.2.84 EF_{SDN} (Service Dial~~l~~ing 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 at the CSIM ADF level. It may also contain associated alpha-tagging.

Identifier: '6F79'	Structure: linear fixed	Optional
Record length: X+14 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1-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	Dial~~l~~ing Number/SSC String	M	10 bytes
X+13	Reserved ('FF') ~~Capability/Configuration2 (EF_{CCP2}) Record Identifier~~	M	1 byte
X+14	Extension3 (EF_{EXT3}) Record Identifier	M	1 byte

For contents and coding of all data items see the respective data items of the EF_{ADN} (Section 5.4.1), with the exception that extension records are stored in the EF_{EXT3} ~~and capability/configuration parameters are stored in EF_{CCP2}~~.

NOTE: The value of X (the number of bytes in the alpha-identifier) may be different ~~to~~ from the length denoted X in EF_{ADN}.

5.2.85 EF_{EXT2}(Extension2)

This EF contains extension data of an FDN (see FDN in 5.2.27).

Identifier: '6F7A'	Structure: linear fixed	Optional
Record length: 13 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE PIN2 DEACTIVATE ADM ACTIVATE 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 Section 5.4.2 (EF_{EXT1}).

5.2.86 EF_{EXT3}(Extension3)

This EF contains extension data of an SDN (see SDN in 5.2.81).

Identifier: '6F7B'	Structure: linear fixed	Optional
Record length: 13 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE ADM DEACTIVATE ADM ACTIVATE 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 Section 5.4.2 (EF_{EXT1}).

5.2.87 EF_{ICI} (Incoming Call Information)

If service n28 is "available", this file shall be present.

This EF is located within the CSIM application. The incoming call information can be linked to the phone book stored under $DF_{TELECOM}$ or to the local phone book within the CSIM. The EF_{ICI} contains the information related to incoming calls.

The time of the call and duration of the call are stored in this EF. This EF can also contain associated alpha identifier that may be supplied with the incoming call. In addition, it contains identifiers of associated network/bearer capabilities and identifiers of extension records at the CSIM ADF level. The structure of this EF is cyclic, so the contents shall be updated only after a call is disconnected.

If Calling Line Identifier is supported and the incoming phone number matches a number stored in the phone book the incoming call information is linked to the corresponding information in the phone book. If the incoming call matches an entry but is indicated as hidden in the phone book the link is established but the information is not displayed by the ME if the code for the secret entry has not been verified. The ME shall not ask for the secret code to be entered at this point.

Optionally the ME may store the link to phone book entry in the file, so that it does not need to look again for a match in the phone book when it reuses the entry. But the ME will have to check that the incoming call number still exits in the linked phone book entry, as the link might be broken (entry modified). When not used by the ME or no link to the phone book has been found, this field shall be set to 'FFFFFF'.

The first byte of this link is used to identify clearly the phone book location either global (i.e. under $DF_{TELECOM}$) or local (i.e. CSIM specific).

For the current version of the phone book, the phone book entry is identified as follows:

- the record number in the EF_{PBR} which indicates the EF_{ADN} containing the entry;
- the record number inside the indicated EF_{ADN}.

The structure of EF_{ICI} is shown below. Coding scheme is according to EF_{ADN}

Identifier: '6F7C'	Structure: Cyclic	Optional
SFI: '10'		
Record length: X+28 bytes	Update activity: high	

Access Conditions:

READ	PIN
UPDATE	PIN
DEACTIVATE	ADM
ACTIVATE	ADM

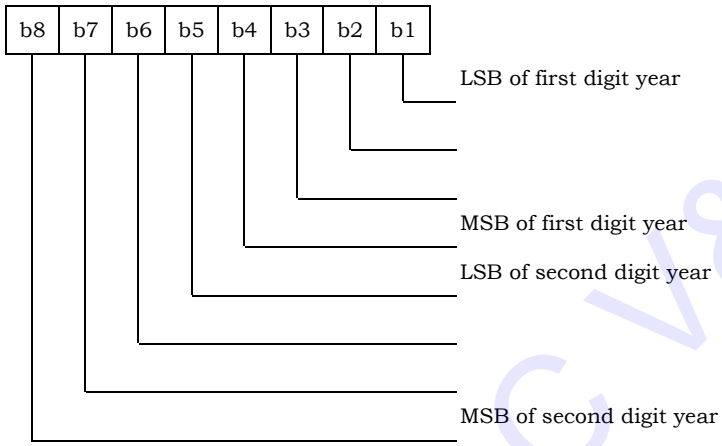
Bytes	Description	M/O	Length
1 to X	Alpha Identifier	O	X bytes
X+1	Length of BCD number contents	M	1 byte
X+2	TON and NPI	M	1 byte
X+3 to X+12	Incoming Call Number	M	10 bytes
X+13	Reserved ('FF')~~Capability/Configuration2 (EF_{CCP2}) Record Identifier~~	M	1 byte
X+14	Extension5 (EF_{EXT5}) Record Identifier	M	1 byte
X+15 to X+21	Incoming call date and time (see detail 1)	M	7 bytes
X+22 to X+24	Incoming call duration (see detail 2)	M	3 bytes
X+25	Incoming call status (see detail 3)	M	1 byte
X+26 to X+28	Link to phone book entry (see detail 4)	M	3 bytes

NOTE: When the contents except incoming call status are invalid, they are filled with 'FF'.

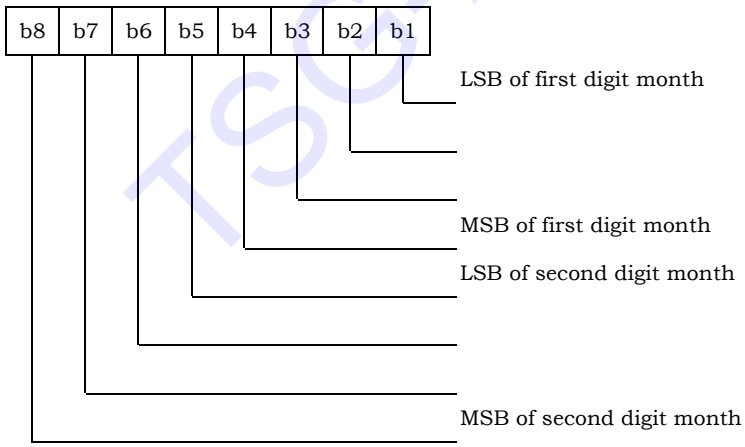
Detail 1: Coding of date and time.

Content:
the date and time are defined by the ME.

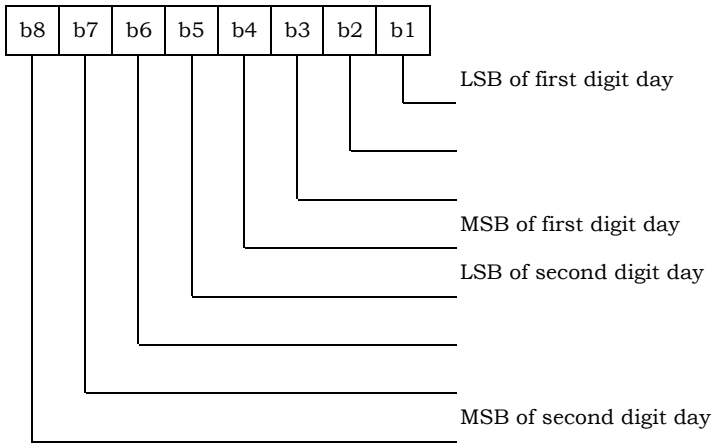
Coding:
it is according to the extended BCD coding from Byte1 to Byte 7. The first 3 bytes show year, month and day (yy.mm.dd). The next 3 bytes show hour, minute and second (hh.mm.ss). The last Byte 7 is Time Zone. The Time Zone indicates the difference, expressed in quarters of an hour, between the local time and GMT. Bit 4 in Byte 7 represents the algebraic sign of this difference (0: positive, 1: negative). If the terminal does not support the Time Zone, Byte 7 shall be "FF". Byte X+15: Year.



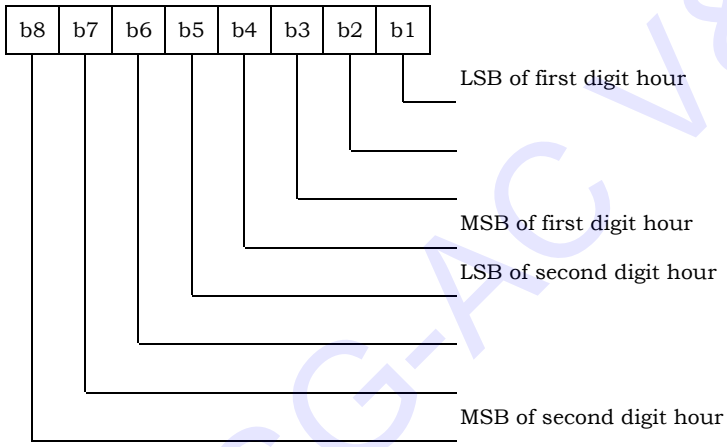
Byte X+16: Month



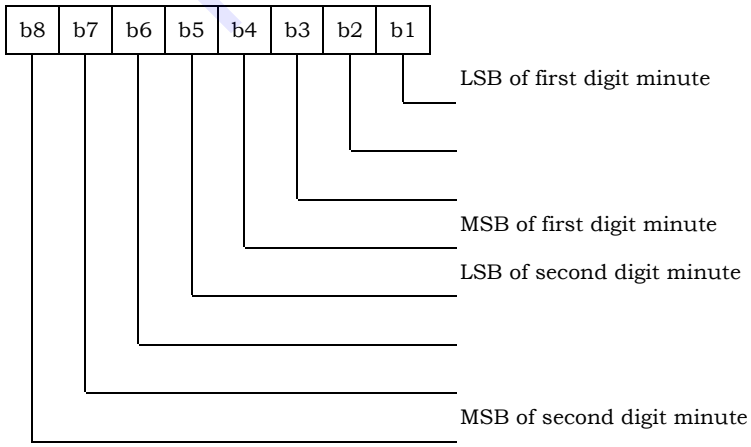
Byte X+17: Day



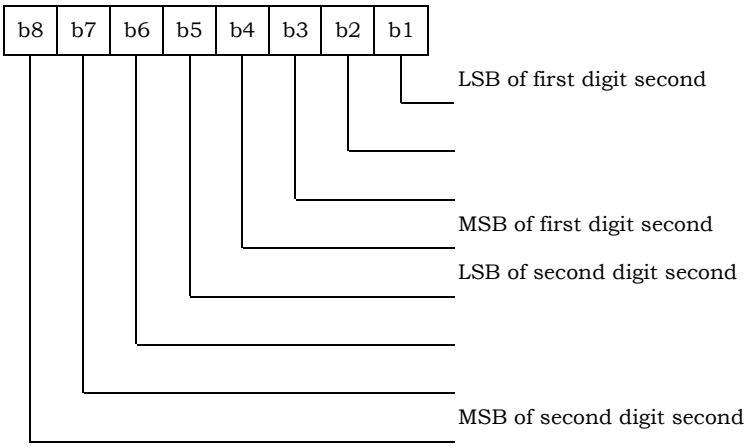
Byte X+18: Hour



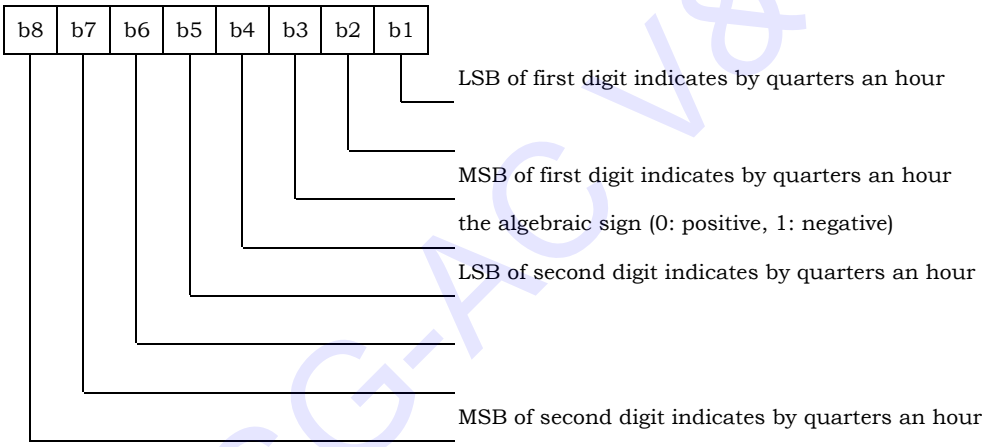
Byte X+19: Minute



Byte X+20: Second



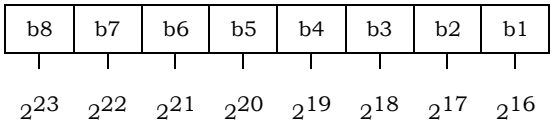
Byte X+21: Time Zone



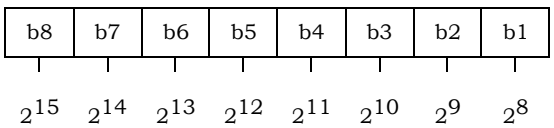
Detail 2: Coding of call duration.

Call duration is indicated by second.

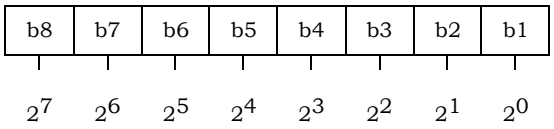
Byte X+22:



Byte X+23:



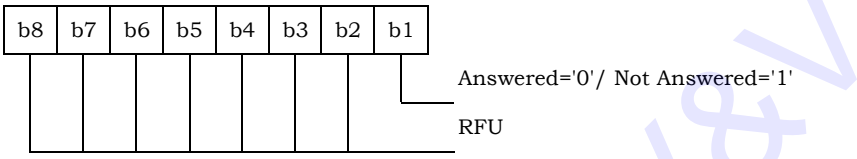
Byte X+24:



For instance, '00' '00' '30' represents 2^5+2^4.

Detail 3: Coding of Call status.

Byte X+25:

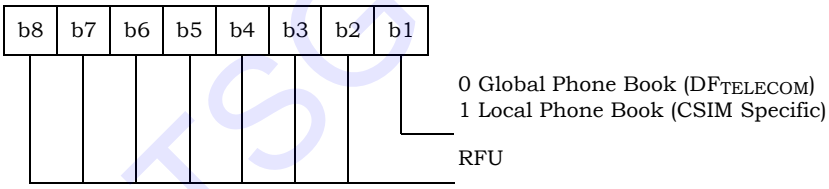


Detail 4: Link to phone book entry

For the current implementation of the phone book the following coding applies:

Phone book reference.

Byte X+26:



EF_{PBR} record number:

Byte X+27: Hexadecimal value.

EF_{ADN} record number:

Byte X+28: Hexadecimal value.

5.2.88 EF_{OCI} (Outgoing Call Information)

If service n27 is "available", this file shall be present.

The outgoing call information can be linked to the phone book stored under $DF_{TELECOM}$ or to the local phone book within the CSIM. The EF_{OCI} contains the information related to outgoing calls.

The time of the call and duration of the call are stored in this EF. It may also contain associated alpha identifier. In addition it contains identifiers of associated network/bearer capabilities and identifiers of extension records at the CSIM ADF level. The structure of this file is cyclic, so the contents shall be updated only after a call is disconnected.

If the dialled phone number matches a number stored in the phone book the outgoing call information might be linked to the corresponding information in the phone book. The dialled number may match with a hidden entry in the phone book. If the dialled number matches a hidden entry in the phone book the link is established but the information related to the phone book entry is not displayed by the ME, if the hidden code has not been verified. The ME shall not perform hidden code verification at this point.

Optionally, the ME may store the link to phone book entry in the file, so that it does not need to look again for a match in the phone book when it reuses the entry. But the ME will have to check that the outgoing call number still exists in the linked phone book entry, as the link might be broken (entry modified). When not used by the ME or no link to the phone book has been found, this field shall be set to 'FFFFFF'.

Coding scheme is according to EF_{ICI}.

Identifier: '6F7D'	Structure: Cyclic	Optional
SFI: '11'		
Record length: X+27 bytes	Update activity: high	

Access Conditions:

READ	PIN
UPDATE	PIN
DEACTIVATE	ADM
ACTIVATE	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	Outgoing Call Number/SSC String	M	10 bytes
X+13	Reserved ('FF') Capability/Configuration2 (EF_{CCP2}) Record Identifier	M	1 byte

X+14	Extension5 (EF_{EXT5}) Record Identifier	M	1 byte
X+15 to X+21	Outgoing call date and time	M	7 bytes
X+22 to X+24	Outgoing call duration	M	3 bytes
X+25 to X+27	Link to Phone Book Entry	M	3 bytes

NOTE: When the contents are invalid, they are filled with 'FF'.

5.2.89 EF_{EXT5} (Extension 5)

This EF contains extension data of EF_{ICI} and EF_{OCI} of the CSIM application.

Identifier: '6F7E'	Structure: linear fixed	Optional
Record length: 13 bytes	Update activity: low	
Access Conditions: READ PIN UPDATE PIN DEACTIVATE ADM ACTIVATE 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 Section 5.4.2 (EF_{EXT1}).

5.2.90 EF_{CCP2} (Capability Configuration Parameters 2)

This EF contains ~~parameters of required network and bearer capabilities and terminal configurations associated with a call established using a fixed dialling number, a service dialling number, an incoming call, or an outgoing call. It~~ no valid information ~~is referred by EF_{FDN}, EF_{SDN}, EF_{ICI} and EF_{OCI}, at CSIM ADF level~~.

Identifier: '6F7F'	Structure: linear fixed	Optional
SFI: '12'		
Record length: X bytes, X≥15	Update activity: low	
Access Conditions: READ PIN UPDATE PIN DEACTIVATE ADM ACTIVATE ADM		

Bytes	Description	M/O	Length
1 to X	~~Bearer capability information element~~Reserved (fill with 'FF')	M	X bytes

~~Unused bytes are filled with 'FF'.~~

5.2.91 ~~EF_{ICCID} (ICC Identification)~~Reserved

~~EF_{ICCID} is defined in [18] with the following restrictions:~~

- ~~This EF shall contain 18 digits of the actual ICCID followed by the check digit and a single 0xF filler digit.~~
- ~~The ICCID shall be globally unique, using an Issuer Identifier Number registered with the ITU-T as specified in [58].~~
- ~~If the long form of the EUIMID is chosen, the ICCID is the LF_EUIMID.~~

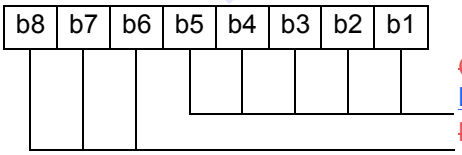
5.2.92 $EF_{AppLabels}$ (Application Labels)

This EF contains text labels that shall be associated with the icons or menu items used to launch applications. ~~These~~ Use of these labels ~~are~~ is optional and need only be provisioned if an operator desires to override the ~~handset vendor~~ME-defined labels.

Identifier: '6F80'	Structure: Transparent	Optional
File size: 4+N*32	Update Activity: Low	
Access Conditions:		
READ	PIN	
UPDATE	ADM	
INVALIDATE	ADM	
REHABILITATE	ADM	

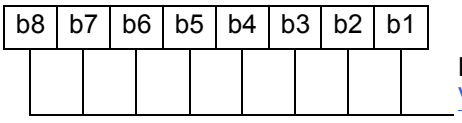
Bytes	Description	M/O	Length
1	Character Encoding	M	1 byte
2	Language Indicator	M	1 byte
3 – 4	Application Labels Present	M	2 bytes
5 – 36	Application $Label_1$	O	32 bytes
37 – 68	Application $Label_2$	O	32 bytes
…	…	O	…
5+(N-1)*32 to 36+(N-1)*32	Application $Label_N$	O	32 bytes

- **Character Encoding:**



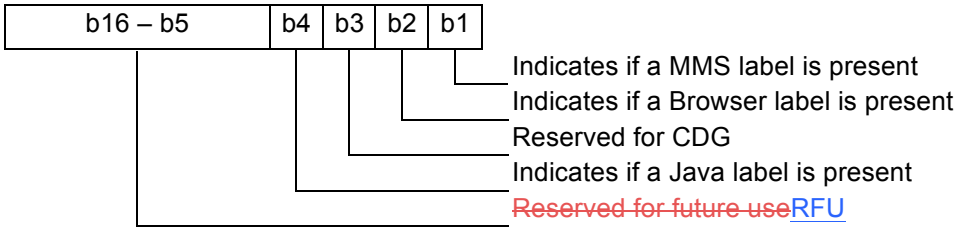
~~Character~~ CHARi encoding type per Table 9.1-1, Data Field Encoding Assignments, of [Informative 1]
~~Reserved for future use~~RFU

- **Language Indicator:**



Language Indicator per Table 9.2-1, Language Indicator Value Assignments of [Informative 1]

- **Application Labels Present:** This field is a bitmask used to identify which Application Label Fields are present in the EF. Each bit represents a particular application as shown below:



If a bit is set to '1,' an Application Label Field for that application shall be present. If the bit is set to '0,' an Application Label Field for that application shall not be present and the ~~handset's~~ ME user interface will display the generic label for that application.

- **Application Label:** Each Application Label field contains the text label to be displayed with the icon or menu item used to launch that application. The Application Label Present field identifies which Application Label fields are present in the EF. These Application Label fields shall be present in the same order as their corresponding bits in the Application Labels Present field. The string contents of each Application Label field shall use the SMS convention as defined in Tables 9.1-1 and 9.2-1 of [Informative 1]. The string shall be left justified. Unused bytes shall be set to 'FF.'

  If the string is coded as 7-bit, the SMS default 7 bit coded alphabet as referenced in [Informative 1] with bit 8 set to 0 shall be used.

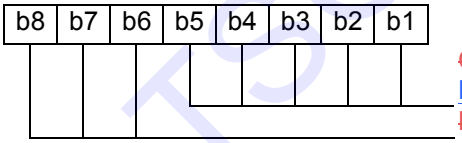
5.2.93 EF_{Model} (Device Model Information)

This EF contains the model information of the ME. Similar to $EF_{ESN\_MEID\_ME}$, this EF is populated by the device during power-up. This EF enables CCAT applications to provide model information to the network either automatically or on demand.

Identifier: '6F81'	Structure: Transparent	Optional
File Size: 126	Update activity: Low	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

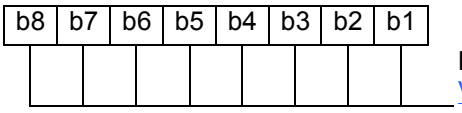
Bytes	Description	M/O	Length
1	Character Encoding	M	1 byte
2	Language Indicator	M	1 byte
3-34	Model Information	M	32 bytes
35-66	Manufacturer Name	M	32 bytes
67-126	Software Version Information	M	60 bytes

- **Character Encoding:**



~~Character~~ CHARi encoding type per Table 9.1-1, Data Field Encoding Assignments, of [Informative 1]
~~Reserved for future use~~RFU

- **Language Indicator:**



Language Indicator per Table 9.2-1, Language Indicator Value Assignments, of [Informative 1]

- **Model Information:** This field is a string indicating the model name of the device (e.g., "ABCCOM-XYZ"). The string contents shall use the SMS convention as defined in Tables 9.1-1 and 9.2-1 of [Informative 1]. The string shall be left justified. Unused bytes shall be set to 'FF.'

- **Manufacturer Name:** This field is a string indicating the manufacturer of the device. The string contents shall use the SMS convention as defined in Tables 9.1-1 and 9.2-1 of [Informative 1]. The string shall be left justified. Unused bytes shall be set to 'FF.'

- **Software Version Information:** This field is a string indicating the software version of the device (e.g., "6.0 patch 01"). The string contents shall use the SMS convention as defined in Tables 9.1-1 and 9.2-1 of [Informative 1]. The string shall be left justified. Unused bytes shall be set to 'FF.'

5.2.94 EF_{RC} (Root Certificates)

If service n36 (Root Certificates) is allocated, this EF shall be present.

This EF contains the root certificates for applications on the device. One or more applications are associated with each certificate.

Identifier: '6F82'	Structure: Transparent	Optional
File Size: X_1+…+X_n	Update activity: Low	
Access Conditions: READ ALW UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to X_1	Certificate TLV Object	M	X_1 bytes
X_1+1 to X_1+X_2	Certificate TLV Object	O	X_2 bytes
…	…	O	…
X_1+…+X_{n-1}+1 to X_1+…+X_n	Certificate TLV Object	O	X_n bytes

Unused bytes shall be set to 'FF.' A Tag value of 'FF' indicates the end of valid data.

- **Certificate TLV Object – Contents:**

Description	Value	M/O	Length
Certificate Tag	'80'	M	1 byte
Length	Note 1	M	Note 2
Certificate Type	Note 3	M	1 byte
Certificate Information	Note 4	M	Variable
Applications	Note 3	M	2 bytes

NOTE 1: This is the total size of the constructed TLV object (not including the tag and this length).

NOTE 2: The length is coded according to [60] using primitive encoding and the minimum number of octets~~ISO/IEC 8825. It is the overall length of data fields after this Length field~~.

NOTE 3: See coding below.

NOTE 4: Binary data for the certificate information as defined in corresponding Certificate Type as defined below, e.g., X.509.

- **Certificate Type – Coding:**

Value	Name	Notes
0	DER Encoded Binary X.509	See section 7 “Public-keys and public-key certificates” in [59] for the definition. The binary encoding is per DER encoding defined in [60].
1	Base64 Encoded X.509	See section 7 “Public-keys and public-key certificates” in [59]. The encoding is per DER encoding defined in [60] and the DER binary data is converted to Base 64 text format.
2	PKCS #7	See section 6.5 “ExtendedCertificateOrCertificate” in [61] for the definition. The binary encoding is per DER encoding defined in [60].
3	PKCS #12	See section 4.2.3 “The CertBag type” in [62] for the definition. The binary encoding is per DER encoding defined in [60].
4-255	Reserved for future use	

- **APPLICATIONS:** This field is a bitmask used to indicate which applications are associated with a particular certificate. If the same certificate is being used for all applications signed by the operator, only bit 1 (Unspecified) will be set. Otherwise, if the operator signs different applications using different certificates, the bit for each application associated with the certificate shall be set. Note that, while each certificate may be associated with multiple applications, each application may only be associated with one certificate.

Bit	Application
1	Unspecified (all applications use the same profile)
2	Reserved
3	WAP Browser
4	Reserved for CDG
5	Java
6	Reserved for CDG
7	Terminal (tethered mode for terminal access)
8-16	Reserved for future use

1

5.2.95 EF_{SMSCAP} (SMS Capabilities)

If services n6 (Short Message Storage) and n35 (Messaging and 3GPD Extensions) are allocated, this EF shall be present.

This EF contains information about SMS Capabilities.

Identifier: '6F83'	Structure: Transparent	Optional
File size: 4 bytes	Update Activity: Low	
Access Conditions:		
READ	PIN	
UPDATE	ADM	
INVALIDATE	ADM	
REHABILITATE	ADM	

Bytes	Description	M/O	Length
1	SMS Retry Period	M	1 byte
2	SMS Retry Interval	M	1 byte
3	SMS Flags	M	1 byte
4	SMS Preferred Service Option	M	1 byte

- **SMS Retry Period:** This is the overall time period (in seconds) during which the Mobile Originated (MO) SMS retries can be performed. 0 means that MO SMS retry is disabled.
- **SMS Retry Interval:** This is the time interval (in seconds) that the device shall wait before the next retry attempt can be made after a MO SMS failure.
- **SMS Flags:** 0 – disabled; 1 – enabled

Bit	Parameter Indicated
1	Send On Access (Allow MO SMS to be sent over Access Channel)
2	Send On Traffic (Allow MO SMS to be sent over Traffic Channel)
3	Send as Standard EMS (Network supports standard EMS per [8])
4-8	~~Reserved for future use~~RFU

- **SMS Preferred Service Option**: This is the preferred service option to be used when the device sets up SMS traffic channel for sending messages.

Value	Description
0	Device Default
1	Service Option 6
2	Service Option 14
3-255	Reserved for future use

5.2.96 $EF_{MIPFlags}$ (Mobile IP Flags)

If services n15 (3GPD-MIP) and n35 (Messaging and 3GPD Extensions) are allocated, this EF shall be present.

This EF contains the configuration flags for Mobile IP.

Identifier: '6F84'	Structure: Transparent	Optional
File size: 1 byte	Update Activity: Low	
Access Conditions:		
READ	PIN	
UPDATE	ADM	
INVALIDATE	ADM	
REHABILITATE	ADM	

Bytes	Description	M/O	Length
1	MIP_FLAGS	M	1 byte

- **MIP_FLAGS:** 0 – disabled; 1 – enabled

Bit	Parameter Indicated
1	Mobile IP ~~2002bis~~ MN HA Authentication. When this bit is enabled MN HA Authentication should be according to [23]. When this bit is disabled MN HA Authentication should be according to [75].
2	Mobile IP Pre Rev 6 handoff optimization
3	Mobile IP PPP Re-sync during hand-down from 1xEV-DO Rev 0 to 1x
4	Mobile IP Re-registration only if data has been transferred since last registration in order to extend Mobile IP address lifetime
5-8	~~Reserved for future use~~RFU

5.2.97 ~~EF$_{SIPUPPExt}$~~ EF$_{3GPDUPPExt}$ (~~SimpleIP~~ 3GPD User Profile Parameters Extension)

If service~~s~~ n14 (3GPD-SIP) or n15 (3GPD-MIP) is allocated and service n35 (Messaging and 3GPD Extensions) ~~are~~ is allocated, this EF shall be present.

This EF contains ~~the~~ additional parameters for Simple IP and Mobile IP User Profiles in order to fully support the feature of multiple profiles.

Identifier: '6F85'	Structure: Transparent	Optional
File size: X bytes	Update Activity: Low	
Access Conditions:		
READ	PIN	
UPDATE	ADM	
INVALIDATE	ADM	
REHABILITATE	ADM	

Bytes	Description	M/O	Length
X	UPP Extension Block	M	X bytes

Unused bytes shall be set to 'FF.'

- **UPP Extension Block structure:**

Field	Length (bits)
NUM_NAI	4
NUM_NAI occurrences of the following fields:	
NAI_ENTRY_INDEX	4
APPLICATIONS	32
PRIORITY	8
DATA_RATE_MODE	4
DATA_BEARER	4
RESERVED	0 or 4

- **NUM_NAI:** Number of UPP Extension instances. This number shall be the same as NUM_NAI in the base user profile EF (EF$_{SIPUPP}$ or EF$_{MIPUPP}$).

- **NAI_ENTRY_INDEX:** Index to the list of UPP Extension instances. This index shall point to the UPP Extension instance that is corresponding to the base UPP instance with the same index value as defined in EF_{SIPUPP} or EF_{MIPUPP}.

- **APPLICATIONS:** This field is a bitmask used to indicate which applications are associated with a particular profile. The applications shall use the profile having the "Unspecified" bit set in the APPLICATIONS bitmask if they are not present in any other profiles.

Bit	Application
1	Unspecified *(used by applications not present in any other profile)*
2	MMS
3	WAP Browser
4	Reserved for CDG
5	Java
6	Reserved for CDG
7	Terminal *(tethered mode for terminal access)*
8	Operator Administration (e.g. BIP)
~~8~~9-32	Reserved for future use

- **PRIORITY:** When attempting to launch a new application, it is possible that another application is already active and has already established a data session. If the new application has the same PRIORITY value as the previous application that established the existing data session, the new application may simply reuse the existing data session.

If the new application has a different PRIORITY than the previous application that set up the existing data session, the device may use the PRIORITY to determine which application has higher priority, as follows:

Value	Priority
0	Highest priority category
1	Second highest priority category (lower than 0; higher than 2 and others)
2	Third highest priority category (lower than 0 or 1; higher than 3 and others)
:	:
255	Lowest priority

- **DATA_RATE_MODE:** Data Rate Mode

Value	Application
0	Low Speed: Low speed service options only
1	Medium Speed: F-SCH with service option 33 only
2	High Speed: F-SCH and R-SCH with service option 33
3-15	Reserved for future use

- **DATA_BEARER**: Data Bearer

Value	Application
0	Hybrid 1x/1xEV-DO
1	1x only
2	1xEV-DO only
3-15	Reserved for future use

5.2.98 Reserved~~EF$_{MIPUPPExt}$ (MobileIP User Profile Parameters Extension)~~

~~If services n15 (3GPD-MIP) and n35 (Messaging and 3GPD Extensions) are allocated, this EF shall be present.~~

~~This EF contains the additional parameters for Mobile IP User Profiles in order to fully support the feature of multiple profiles for Mobile IP.~~

~~Identifier: '6F86'~~	~~Structure: Transparent~~	~~Optional~~
~~File size: X bytes~~	~~Update Activity: Low~~	
~~Access Conditions:~~		
~~READ~~	~~PIN~~	
~~UPDATE~~	~~ADM~~	
~~INVALIDATE~~	~~ADM~~	
~~REHABILITATE~~	~~ADM~~	

~~Bytes~~	~~Description~~	~~M/O~~	~~Length~~
~~X~~	~~UPP Extension Block~~	~~M~~	~~X bytes~~

~~The UPP Extension Block is used by both EF$_{SIPUPPExt}$ for Simple IP and EF$_{MIPUPPExt}$ for Mobile IP. See the definition of EF$_{SIPUPPExt}$ for the definition of the UPP Extension Block.~~

5.2.99 $EF_{IPV6CAP}$ (IPv6 Capabilities)

If services n35 (Messaging and 3GPD Extensions) and n41 (IPv6) are allocated, this EF shall be present.

This EF contains information about IPv6 capabilities.

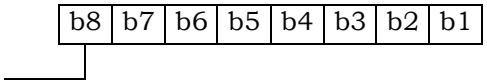
Identifier: '6F87'	Structure: Transparent	Optional
File size: 21 bytes	Update Activity: Low	
Access Conditions:		
READ	PIN	
UPDATE	ADM	
INVALIDATE	ADM	
REHABILITATE	ADM	

Bytes	Description	M/O	Length
1-2	Initial neighbor solicitation delay time	M	2 bytes
3-4	Solicitation interval	M	2 bytes
5-6	Re-solicitation interval	M	2 bytes
7-8	Maximum solicitation attempts	M	2 bytes
9-10	Maximum re-solicitation attempts	M	2 bytes
11-12	Pre-RA expiry re-solicitation time	M	2 bytes
13-20	IID Information	M	8 bytes
21	IPv6 Flags	M	1 byte

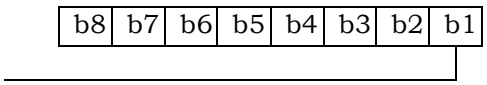
- **Initial neighbor solicitation delay time** *(in units of 100ms)*: Time ~~mobile~~ MS waits after the IID (Interface ID) has been negotiated before sending an RS (Router Solicitation) in an attempt to receive an RA (Router Advertisement).

Coding: 16-bit integer.

Byte 1:



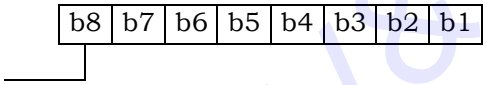
Byte 2:



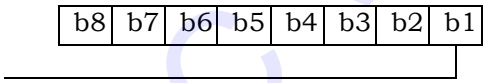
- **Solicitation interval** *(in units of 100ms)***:** Amount of time the ~~mobile~~ MS waits before sending a subsequent RS after a previous one.

  Coding: 16-bit integer.

  Byte 1:



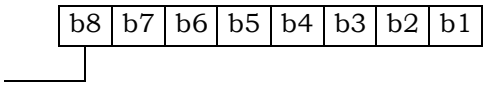
  Byte 2:



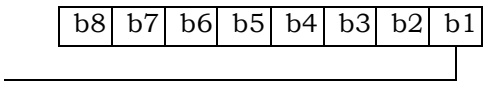
- **Re-solicitation interval** *(in units of 100ms)***:** Amount of time between solicitations sent while re-soliciting for a new RA. This interval applies only after the ~~mobile~~ MS has previously received one valid RA and is soliciting for a new one to renew the lifetimes of the current prefix or retrieve a non-deprecated prefix.

  Coding: 16-bit integer.

  Byte 1:



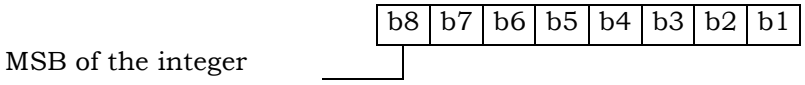
  Byte 2:



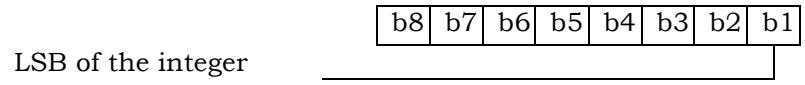
- **Max solicitation attempts:** Number of solicitation attempts to make for initial IPv6 session establishment, when an RA is not received in response before giving up IPv6 auto-configuration.

  Coding: 16-bit integer.

  Byte 1:

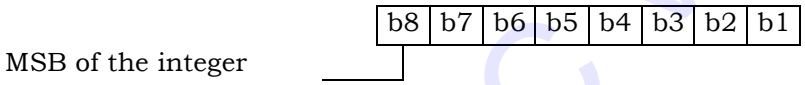
  Byte 2:

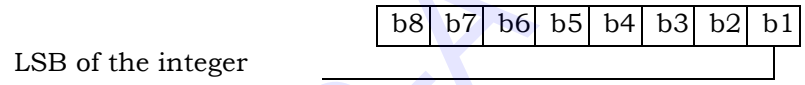
- **Max re-solicitation attempts:** Number of solicitation attempts to make to re-solicit for a new RA.

  Coding: 16-bit integer.

  Byte 1:

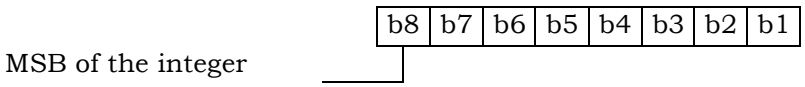
  Byte 2:

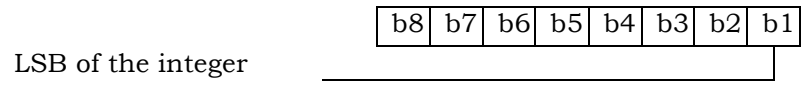
- **Pre-RA expiry re-solicitation time** *(in units of 100ms)***:** Amount of time before the current RA expires to begin re-solicitations.

  Coding: 16-bit integer.

  Byte 1:

  Byte 2:

- **IID Information:** IID is part of the IPv6 address. See [63] for information on coding.

- **IPv6 Flags:** Identify IPv6 behavior. Coding (0 – Disabled; 1 – Enabled).

Bit	Parameter Indicated
1	Use IPv6
2	Failover from IPv6 to IPv4
3	PDSN as proxy IPv6 DNS server. When enabled, the ~~mobile~~ MS forwards all DNS requests to the PDSN. The PDSN forwards requests to the appropriate DNS server. This parameter is meaningful only if the primary and secondary DNS server addresses are not available.
4-8	~~Reserved for future use~~RFU

5.2.100 $EF_{TCPConfig}$ (TCP Configurations)

If service n14 (3GPD-SIP) or n15 (3GPD-MIP) is allocated and service n35 (Messaging and 3GPD Extensions) is allocated, this EF shall be present.

This EF contains information about Transmission Control Protocol configurations.

Identifier: '6F88'	Structure: Transparent	Optional
File size: 2 bytes	Update Activity: Medium	
Access Conditions:		
READ	PIN	
UPDATE	ADM	
INVALIDATE	ADM	
REHABILITATE	ADM	

Bytes	Description	M/O	Length
1	TCP Flags	M	1 byte
2	TCP Keep-Alive Idle Timer	M	1 byte

- **TCP Flags:** Coding (0 – Disabled; 1 – Enabled):

Bit	Parameter Indicated
1	TCP Graceful close of dormant connections
2-8	~~Reserved for future use~~RFU

- **TCP Keep-Alive Idle Timer:** Coding: Number of minutes. A value of 0 means that the TCP keep-alive feature is disabled on the ME.

5.2.101 EF_{DGC} (Data Generic Configurations)

If service n14 (3GPD-SIP) or n15 (3GPD-MIP) is allocated and service n35 (Messaging and 3GPD Extensions) is allocated, this EF shall be present.

This EF contains miscellaneous data configuration items.

Identifier: '6F89'	Structure: Transparent	Optional
File size: 3 bytes	Update Activity: Medium	
Access Conditions:		
READ	PIN	
UPDATE	ADM	
INVALIDATE	ADM	
REHABILITATE	ADM	

Bytes	Description	M/O	Length
1	Data dormant timer	M	1 byte
2	EPZID Type Information	M	1 byte
3	Hysteresis Activation Time	M	1 byte

- **Data dormant timer:** Number of seconds to wait before going into data dormant mode, which shall be at least 20 seconds.
- **EPZID Type Information:** Contains the Extended Packet Zone ID Types.

Value	Description
0	Packet Zone ID
1	Packet Zone ID plus SID
2	Packet Zone ID plus SID and NID
3-255	Reserved for future use

- **Hysteresis Activation Time:** This is the number of seconds that the device should wait before it goes into hysteresis state and adds new Packet Zone IDs to the packet zone list as needed. See [65] for details on usage of this timer.

5.2.102 $EF_{WAPBrowserCP}$ (WAP Browser Connectivity Parameters)

If service n37 (WAP Browser) is allocated, this EF shall be present.

This EF contains the connectivity parameters for a WAP Browser application, such as Gateway and Home URL information. At least one gateway shall be configured in this EF as the primary gateway for browsing. Additional gateways as part of the additional instances of Connectivity Parameters can be optionally configured as secondary gateways in the order of priority as they appear in this EF.

Identifier: '6F8A'	Structure: Transparent	Optional
File Size: X_1+…+ X_n	Update activity: Low	
Access Conditions: READ PIN UPDATE ADM INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to X_1	WAP Browser Connectivity Parameters TLV object	M	X_1 bytes
X_1+1 to X_1 + X_2	WAP Browser Connectivity Parameters TLV object	O	X_2 bytes
…	…		
X_1+…+ X_{n-1}+1 to X_1+…+ X_n	WAP Browser Connectivity Parameters TLV object	O	X_n bytes

Unused bytes shall be set to 'FF.' A Tag value of 'FF' indicates the end of valid data.

- **WAP Browser Connectivity Parameters Tags:**

Description	Tag Value
WAP Browser Connectivity Parameters Tag	'AC'
Gateway Tag	'83'
HomeURL Tag	'80'

- **WAP Browser Connectivity Parameters TLV Object contents:**

Description	Value	M/O	Length (bytes)
WAP Browser Connectivity Parameters Tag	'AC'	M	1
Length	Note 1	M	Note ~~1~~2
Gateway Tag	'83'	O	1
Gateway Length	Z	O	Note 2
Gateway Information	--	O	Z
HomeURL Tag	'80'	M	1
HomeURL Length	X	M	Note 2
HomeURL Information	--	M	X
NOTE 1: This is the total size of the constructed TLV object (not including the tag and this length). NOTE 2: The length is coded according to [60] using primitive encoding and the minimum number of octets~~ISO/IEC 8825~~.			

- **Gateway Tag:** This contains information needed to access the WAP Gateway/Proxy server. See description of EF_{MMSICP} for the definition of Gateway TLV Object.

- **HomeURL Tag:** This contains the URL for the WAP Browser's home page for the current particular connectivity parameters. For contents and syntax of URL TLV data object values, see [~~52~~64]. The URL shall be encoded to an octet string according to UTF-8 encoding rules as specified in [~~46~~67].

5.2.103 $EF_{WAPBrowserBM}$ (WAP Browser Bookmarks)

If service n37 (WAP Browser) is allocated, this EF shall be present.

This EF contains bookmarks that may be provisioned by the operator and/or updated by the user.

Identifier: '6F8B'	Structure: Transparent	Optional
File Size: Variable	Update activity: High	
Access Conditions: READ PIN UPDATE PIN INVALIDATE ADM REHABILITATE ADM		

Bytes	Description	M/O	Length
1 to X_1	Bookmark TLV object	M	X_1 bytes
X_1+1 to X_1+X_2	Bookmark TLV Object	O	X_2 bytes
…	…	O	…
X_1+X_2+…+X_{n-1}+1 to X_1+X_2+…+X_{n-1}+X_n	Bookmark TLV Object	O	X_n bytes

Unused bytes shall be set to 'FF.' A value of 'FF' in place of Bookmark Tag field indicates the end of valid data.

- **Bookmark TLV object contents:**

Description	Value	M/O	Length (bytes)
Bookmark Tag	'AD'	M	1
Length	Note 1	M	Note 2
URL Tag	'80'	M	1
Length	Y	M	Note 2
URL Information	--	M	Y
Bookmark Name Tag	'81'	O	1
Length	Z	O	Note 2
Bookmark Name Information	--	O	Z
NOTE 1: This is the total size of the constructed TLV object object (not including the			

tag and this length).
NOTE 2: The length is coded according to [60] using primitive encoding and the minimum number of octets~~ISO/IEC 8825~~.

- **URL Information:** For contents and syntax of URL TLV data object values, see [~~52~~64]. The URL shall be encoded to an octet string according to UTF-8 encoding rules, as specified in [~~46~~67].

- **Bookmark Name Information:** This field shall be encoded to an octet string according to UTF-8 encoding rules as specified in [~~46~~67].

5.2.104 $EF_{MMSConfig}$ (MMS Configuration)

If services n19 (Multimedia Messaging Service) and n35 (Messaging and 3GPD Extensions) are allocated, this EF shall be present.

This EF contains the configuration of MMS.

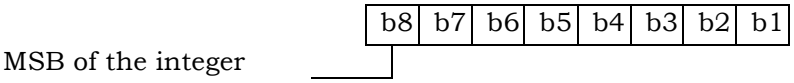
Note that this EF does not contain configuration associated with how the MMS client connects to the MMS service. This type of configuration information is included in ~~the MMS Issuer Connectivity Parameters EF (~~EF_{MMSICP}~~)~~.

Identifier: '6F8C'	Structure: Transparent	Optional
File size: 8 bytes	Update Activity: Medium	
Access Conditions:		
READ	PIN	
UPDATE	ADM	
INVALIDATE	ADM	
REHABILITATE	ADM	

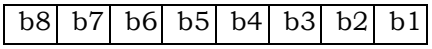
Bytes	Description	M/O	Length
1-4	Max Message Size Value	M	4 bytes
5	Retry Times Value	M	1 byte
6	Retry Interval Value	M	1 byte
7-8	MMSC Timeout Value	M	2 bytes

- **Max Message Size:** This is the maximum MMS message size (in bytes) allowed by the operator. Coding: 32-bit integer.

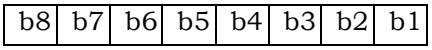
  Byte 1:

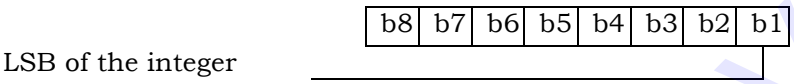
  Byte 2:

  Byte 3:

  Byte 4:

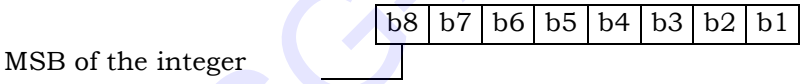
- **Retry Times:** This is the number of times the MMS application will retry for sending a message. Coding: 8-bit integer.

- **Retry Interval:** This is the number of seconds to wait before the next retry is attempted. Coding: 8-bit integer.

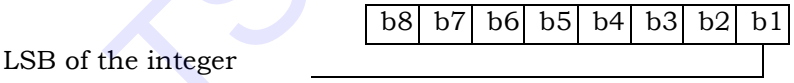
- **MMSC Timeout:** This is the number of seconds for the device to wait for response from Mobile Messaging Service Center (MMSC) before declaring it as an MMSC timeout.

  Coding: 16-bit integer.

  Byte 1:

  Byte 2:

5.2.105 EF_{JDL} (Java Download URL)

If service n38 (Java) is allocated, this EF shall be present.

This EF contains the information for downloading Java applications from the Java download server.

Identifier: '6F8D'		Structure: Transparent	Optional
File size: ~~Variable~~Y≥X bytes		Update Activity: Low	
Access Conditions:			
READ	PIN		
UPDATE	ADM		
INVALIDATE	ADM		
REHABILITATE	ADM		
Bytes	Description	M/O	Length
1-X	Java Download URL	M	X bytes

Unused bytes shall be set to ~~'FF~~'FF.'

- **Java Download URL:** This field of X bytes contains the URL for the Java download server and a termination byte. For contents and syntax, see [~~52~~64]. The URL shall be encoded ~~to~~ as an octet string according to UTF-8 encoding rules, as specified in [~~46~~67]. ~~This string shall be NULL terminated~~The termination byte is set to '00'.

5.2.106 EF_{ARR} (Access Rule Reference)

This EF contains the access rules for files located under the CSIM ADF in the UICC. For the EF structure and description refer to Section 4.2.55 of [30].

5.3 Contents of DFs at the CSIM ADF (Application DF) level

DFs may be present as child directories of CSIM ADF. For this revision, the following DF is defined:

- $DF_{PHONEBOOK}$ '5F3A'.

~~(DF for application specific phonebook. This DF has the same structure as the $DF_{PHONEBOOK}$ under $DF_{TELECOM}$).~~

Note: The $DF_{PHONEBOOK}$ under CSIM ADF (DF for application specific phonebook) has the same structure as the $DF_{PHONEBOOK}$ under $DF_{TELECOM}$.

5.3.1 Contents of files at the $DF_{PHONEBOOK}$ level

The $DF_{PHONEBOOK}$ for CSIM shall comply with all requirements specified in [30] Section 4.4.2, with a restriction that SFI shall not apply to the CSIM. In the context of 3GPP2 systems, "USIM" and "SIM" shall be interpreted as "CSIM" and "R-UIM" respectively.

5.4 Contents of EFs at the $DF_{TELECOM}$ level

5.4.1 EF_{ADN} (Abbreviated dialling numbers)

~~In case of a present~~If DF_{CDMA} [46] is present on the UICC, the first EF_{ADN} (i.e. reflected by the first record in EF_{PBR}) of the $DF_{PHONEBOOK}$ is mapped to the EF_{ADN} under $DF_{TELECOM}$ (with an identifier equal to '6F3A') ~~to $DF_{TELECOM}$~~ to ensure backwards compatibility.

An ME shall not access this file. The information is accessible for the ME in EF_{ADN} under $DF_{PHONEBOOK}$.

5.4.2 EF_{EXT1} (Extension 1)

~~In case of a present~~If DF_{CDMA} [46] is present on the UICC, the first EF_{EXT1} (i.e. reflected by the first record in EF_{PBR}) of the $DF_{PHONEBOOK}$ is mapped to the EF_{EXT1} under $DF_{TELECOM}$ (with an identifier equal to '6F4A') ~~to $DF_{TELECOM}$~~ to ensure backwards compatibility.

An ME shall not access this file. The information is accessible for the ME in EF_{EXT1} under $DF_{PHONEBOOK}$.

5.4.3 EF_{ECCP}, EF_{CCP1}, EF_{CCP} ~~(Extended Capability Configuration Parameter)~~

~~In case of a present~~If DF_{CDMA} ~~application~~ is present on the UICC, the first EF_{CCP1} (i.e. reflected by the first record in EF_{PBR}) of the $DF_{PHONEBOOK}$ under DF_{CDMA} is mapped to the EF_{CCP1} under $DF_{TELECOM}$ (with an identifier equal to '6F4F') ~~to $DF_{TELECOM}$~~ to ensure backwards compatibility. There shall not be any EF_{CCP} (with a file-id of '6F3D') under $DF_{TELECOM}$ because otherwise a R-UIM ME could create inconsistencies within the phonebook.

An ME shall not access this file. The information is accessible for the ME in EF_{CCP1} under $DF_{PHONEBOOK}$.

5.4.4 EF_{SUME} (Set Up Menu Elements)

This File is defined in [54], and has the file identifier '6F54'.

5.4.5 EF_{ARR} (Access Rule Reference)

This EF contains the access rules for files located under the $DF_{TELECOM}$ in the UICC. If the security attribute tag '8B' is indicated in the FCP it contains a reference to a record in this file.

This EF contains one or more records containing access rule information according to the reference to expanded format as defined in [53]. Each record represents an access rule. Unused bytes in the record are set to 'FF'.

If the card cannot access EF_{ARR}, any attempt to access a file with access rules indicated in this EF_{ARR} shall not be granted.

5.5 Contents of DFs at the $DF_{TELECOM}$ level

DFs may be present as child directories of $DF_{TELECOM}$. The following DFs have been defined:

- $DF_{GRAPHICS}$ '5F50'.
- $DF_{PHONEBOOK}$ '5F3A'.

(DF for public phone book. This DF has the same structure as $DF_{PHONEBOOK}$ under ADF CSIM).

- $DF_{MULTIMEDIA}$ '5F3B'.
- DF_{MMSS} '5F3C'.

The DFs and EFs under $DF_{TELECOM}$ are defined in [45] and [30]. The files defined under DF_{MMSS} (file identifier '5F3C') in $DF_{TELECOM}$ are used by CSIM for MMSS support and defined in [45]. This DF shall be present if the card supports MMSS.

5.5.1 Contents of files at the $DF_{GRAPHICS}$ level

The $DF_{GRAPHICS}$ for CSIM shall comply with all requirements specified in [30] Section 4.6.1.

5.5.2 Contents of files at the $DF_{PHONEBOOK}$ under the $DF_{TELECOM}$

This DF has the same structure as $DF_{PHONEBOOK}$ under the ADF_{CSIM}.

5.5.3 Contents of files at the $DF_{MULTIMEDIA}$ level

The EFs in the $DF_{MULTIMEDIA}$ contain multimedia information. This DF shall be present if service n30 is available, i.e. if the card supports MMS storage.

The EFs in the $DF_{MULTIMEDIA}$ for CSIM shall comply with all requirements specified in ~~[30]~~ Section 4.6.3.1 of [30] for EF_{MML} (Multimedia Messages List) and 4.6.3.2 of [30] for EF_{MMDF} (Multimedia Messages Data File). In the context of 3GPP2 systems, reference to [Informative 2] and [Informative 3] shall be interpreted as a reference to [45] and [37] respectively.

5.5.4 Contents of files at the DF_{MMSS} level

The EFs in the DF_{MMSS} contain multimode system selection parameters and settings. The following EFs which support MMSS are defined in Section 4.1 [45].

- EF_{MLPL} – File identifier is '4F20'.
- EF_{MSPL} – File identifier is '4F21'.
- $EF_{MMSSMODE}$ – File identifier is '4F22'.

These EFs can be queried and updated using the CSIM commands defined in section 9.4.2 (OTASP/OTAPA-related Commands).

6. INTERWORKING OF R-UIM & CSIM APPLICATION ON A UICC

An R-UIM [46] and a CSIM implemented together on a single UICC can never be activated at the same time. Neither can they be switched from one to the other. Their activities solely depend on the functionality of ME in which they are inserted: an ME supporting the CSIM shall use the CSIM rather than the R-UIM.

However, both applications may share certain elements to optimize memory consumption, but still, both applications have to be virtually independent from the functional point of view. The following section describes the possible options.

6.1 File Mapping

Many files of R-UIM [46] and CSIM not only have the same name and file identifier (although under different DFs) but are entirely equal by size and content parameters. This generally allows for memory efficient implementation of a CSIM together with an R-UIM, as these files can be shared by both applications, i.e. necessary storage capacity is only required once. Further, shared files speeds up the pre-personalization process as they save valuable programming time.

Therefore, files should be mapped as far as possible, i.e. in all cases where basic properties are equal and identical contents do not conflict with the access by either an R-UIM or a CSIM based ME or with intended subscription differences when separate IMSIs are used.

Annex A gives an overview of the rules for mapping files between an R-UIM and CSIM. A case by case decision should be conducted by the network operator / card manufacturer for each UICC implementation.

Caution: It should be noted that file identifiers may differ between the R-UIM and CSIM, while all other file properties are exactly the same.

6.2 Reserved

6.3 Access conditions

If an EF is accessible in both CSIM and R-UIM operation modes, independent UICC and non-UICC access conditions may be defined for the file. The UICC does not check the consistency of the access conditions in both modes.

Therefore, it is possible that the same EF has different security attributes in UICC and non-UICC operation modes. It is the responsibility of the network operator and the card manufacturer to ensure at the personalization stage that the security attributes for a UICC and non-UICC session are the same, if necessary.

6.4 Reserved

7. APPLICATION PROTOCOL

The requirements stated in the corresponding section of [45] apply to the CSIM application.

The procedures listed in Section 7.1, "CSIM management procedures," are required for execution of the procedures in the Section 7.2, "CSIM security related procedures," and Section 7.3, "Subscription Related Procedures". The procedures listed in Section 7.2, "CSIM security related procedures," are mandatory. The procedures listed in Section 7.3, are only executable if the associated services, which are optional, are provided in the CSIM. However, if the procedures are implemented, it shall be in accordance with Section 7.3. Section 7.4 describes CCAT related procedures.

7.1 CSIM management procedures

If a CSIM application is present on the UICC, an ME shall only use the CSIM application. In this case, a possibly existing R-UIM shall never be used by an ME.

7.1.1 Initialization

7.1.1.1 CSIM Application Selection

After UICC activation (see [45]), the ME selects a CSIM application. If no EF$_{DIR}$ file is found or no CSIM applications are listed in the EF$_{DIR}$ file, the ME may then try to select the R-UIM as specified in [46]. After a successful CSIM application selection, it is the UICC's responsibility to store the selected CSIM ~~(~~AID~~)~~ on the UICC. This application is referred to as the last selected CSIM application. The last selected CSIM application shall be available on the UICC after a deactivation followed by an activation of the UICC.

If a CSIM application is selected using a partial DF name, the partial DF name supplied in the command shall uniquely identify a CSIM application. Furthermore if a CSIM application is selected using a partial DF name as specified in [45] indicating in the SELECT command the last occurrence, the UICC shall select the CSIM application stored as the last CSIM application. If, in the SELECT command, the options first, next/previous are indicated, they have no meaning if an application has not been previously selected in the same session and shall return an appropriate error code.

7.1.1.2 CSIM Initialization

If EF$_{ME3GPDOPC}$ is present, after the selection of CSIM Application, the CSIM shall set the value of Octet 1 in EF$_{ME3GPDOPC}$ to '00'.

The ME performs the Emergency Call Codes request.

The ME performs the Preferred Language request.~~The CSIM application shall not indicate any language preference. It shall use the language indicated by any other application currently active on the UICC or by default, choose a language from EF$_{PL}$ at the MF level according the procedure defined in [45].~~

~~If the ME does not support the languages of EF$_{PL}$, then the ME shall use its own internal default selection.~~

The ME then runs the user verification procedure as defined in Section 6.4 of [30] (where each instance of USIM is replaced with CSIM and where the disabling of PIN2 is always allowed). If the procedure is not performed successfully, the CSIM initialization stops.

~~Then the~~The ME performs the ~~administrative~~ Administrative Data ~~information~~ request.

The ME performs the CSIM Service Table request.

The ME performs the Enabled Services Table request.

The ME performs the OTASP/OTAPA Features request. [7]

~~The ME reads the Administrative Data.~~

The ME ~~reads~~ performs the ~~Removable UIM_ID~~RUIMID request.

The ME ~~sends~~ performs the "~~Store_~~ESN_MEID_ME update"~~ command~~.

The ME performs the ME-specific Configuration Request update.

~~If all these procedures have been performed successfully then CSIM session shall start. In all other cases CSIM session shall not start.~~

~~Afterwards, the~~The ME ~~runs~~ shall run the following procedures if the ME and the CSIM support the related services:

- Service Preferences request;
- AKA (3GCIK) request;
- IMSI ~~Request~~request;
- Access Overload Class information request;
- PRL and EPRL request;
- ~~Preferred Roaming List request;~~
- PUZL request;
- ~~Preferred User Zone List request;~~
- 3GPD Operation Capabilities update;
- Device Model update;
- Multimode Location Association Priority List (MLPL) request;
- Multimode System Priority List (MSPL) request;
- ~~Depending on the further~~Reading of additional EFs depending on the additional services that are supported by both the ME and ~~the~~ CSIM ~~the corresponding EFs have to be read~~.

[7] The OTASP/OTAPA features request is needed to determine which of the OTASP/OTAPA features and feature protocol revisions are supported by the card. This, ~~is~~ in turn, helps the ME determine which of the subsequent OTASP/OTAPA-related requests (e.g. PRL, PUZL) are needed.

After the CSIM initialization has been completed successfully, ~~then~~ the ME is ready for a CSIM session and shall indicate this to the CSIM by sending a ~~particular~~ STATUS command with P1 = '01' (current application is initialized) as defined in section 11.1.2 of [18].

7.1.2 Session Termination

NOTE ~~1~~: This procedure is not to be confused with the deactivation procedure in defined in [45].

The ME shall indicate to the CSIM by sending a ~~particular~~ STATUS command with P1 = '02' as defined in section 11.1.2 of [18] that the termination procedure is starting.

The ME then ~~runs all the procedures which are necessary to transfer the following subscriber related information to the CSIM:~~

~~Key update.~~

~~Finally, the ME~~ deletes all ~~these~~ subscriber related information elements from its memory.

To actually terminate the session, the ME shall then use one of the mechanisms described in Sec. 8.5.3 of [18]~~[45]~~.

7.1.3 CSIM Application Closure

After termination of the CSIM application session as defined in 7.1.2, the CSIM application may be closed by closing the logical channels that are used to communicate with this particular CSIM application.

7.1.4 Emergency call codes request

Request: If EF_{ECC} is present and if the ME supports ECC, the ~~The~~ ME performs the reading procedure with EF_{ECC}. If the user dials a number that matches one of the codes in EF_{ECC}, then the ME shall treat the call as an emergency call as specified in [5]. ~~If EF_{ECC} does not contain any valid number, the ME shall use the emergency numbers it stores for use in setting up an emergency call without a CSIM application.~~

~~Update: The ME performs the updating procedure with EF_{ECC}.~~

~~NOTE: The update procedure is only applicable when the access condition of ADM for "UPDATE" is set to ALW, PIN or PIN2.~~

7.1.5 Preferred Language request~~Language indication~~

The CSIM application shall not indicate any language preference. The ME shall use the language indicated by any other application currently active on the UICC or by default, choose a language from EF_{PL} at the MF level according the procedure defined in [45].

If the ME does not support the languages of EF_{PL}, then the ME shall use its own internal default selection.

~~Request: The ME performs the reading procedure with EF_{LI}.~~

~~Update: The ME performs the updating procedure with EF_{LI}.~~

7.1.6 Administrative ~~information~~ Data request

The ME performs the reading procedure with EF_{AD} and processes the data as appropriate~~.~~

7.1.7 CSIM Service Table request

The ME performs the reading procedure with EF_{CSIM_ST} and processes the data as appropriate.

7.1.8 UICC Presence Detection

The ME checks for the presence of the UICC according to section 14.5.2 of [18] within any period of inactivity no greater than 30 seconds on the UICC-ME interface during a call. If the presence detection fails, the ME shall terminate the call within 5 seconds after the presence detection fails. Here a call includes a circuit switched call or an active packet data call.

7.1.9 Enabled Services Table request

If Service n32 is available, the ME performs the reading procedure with EF_{EST} and processes the data as appropriate.

7.1.10 OTASP/OTAPA Features request

If the ME supports OTASP/OTAPA, the ME performs the reading procedure with EF_{OTA} and processes the data as appropriate.

7.1.11 RUIMID request

The ME performs the reading procedure with EF_{RUIMID} and processes the data as appropriate.

7.1.12 ESN_MEID_ME update

The ME sends the "STORE ESN_MEID_ME" command (see section 9.4.3.1), transferring its ESN_MEID_ME to the UICC and processes the response as appropriate.

7.1.13 ME-specific Configuration Request update

The ME updates the ME-specific Configuration Request Parameters in EF_{MECRP}.

7.1.14 Service Preferences request

The ME performs the reading procedure with EF_{SP} and processes the data as appropriate.

7.1.15 IMSI request

The ME performs the reading procedure with EF_{IMSI_M} and EF_{IMSI_T}.

If IMSI_M_PROGRAMMED is set to '1', then the ME processes the data as appropriate.

If IMSI_T_PROGRAMMED is set to '1', then the ME processes the data as appropriate.

7.1.16 Access Overload Class information request

The ME performs the reading procedure with EF_{ACCOLC} and processes the data as appropriate.

7.1.17 PRL and EPRL request

If the ME supports only SSPR_P_REV= 1, then the ME performs the reading procedure with EF_{PRL} and processes the data as appropriate. Otherwise, if the ME supports SSPR_P_REV ≥ 3, then if EF_{EPRL} is present, then the ME performs the reading procedure with EF_{EPRL} and processes the data as appropriate. Otherwise, if a functional EF_{EPRL} is not present, then the ME performs the reading procedure with EF_{PRL} and processes the data as appropriate.

7.1.18 PUZL request

If the ME supports PUZL_P_REV ≥ 2 and if EF_{PUZL} is present, then the ME performs the reading procedure with EF_{PUZL} and processes the data as appropriate.

7.1.19 3GPD Operation Capabilities update

If services n14 or n15 are available for 3GPD Operation Capabilities, the ME:

- Sets the bit flag for SimpleIP in Octet 1 of $EF_{ME3GPDOPC}$ to '1' if the ME supports SimpleIP.
- Sets the bit flag for MobileIP in Octet 1 of $EF_{ME3GPDOPC}$ to '1' if the ME supports MobileIP.
- Sets the bit flag for MobileIP with SimpleIP Fallback in Octet 1 of $EF_{ME3GPDOPC}$ to '1' if the ME supports MobileIP with SimpleIP Fallback.

7.1.20 Device Model update

If EF_{Model} (Device Model Information) is present, the ME updates the model information in EF_{Model}.

7.1.21 Multimode Location Association Priority List (MLPL) request

If the ME supports MMSS_P_REV ≥ 1 and if EF_{MLPL} is present, then the ME performs the reading procedure with EF_{MLPL} and processes the data as appropriate.

7.1.22 Multimode System Priority List (MSPL) request

If the ME supports MMSS_P_REV ≥ 1 and if EF_{MSPL} is present, then the ME performs the reading procedure with EF_{MSPL} and processes the data as appropriate.

7.2 CSIM Security Related Procedures

All the security related procedures defined in [46] ~~is~~ are applicable to ~~this~~ the CSIM application.

7.2.1 AKA (3GCIK)

Requirement: Service n16 "available".

Request: If the ME supports AKA, then the ME performs the reading procedure with EF_{3GCIK} and restores the CK and IK from the CSIM to the ME per section 4.11.5 of [46].

~~Update: If the ME supports AKA, then the ME performs the updating procedure to store CK and IK in EF_{3GCIK}.~~

7.3 Subscription Related Procedures

7.3.1 Phone book procedure

The Phone book procedures for CSIM shall comply with all requirements specified in [30] Section 5.3.1.

7.3.2 Dialing numbers

Requirements:

- Service n1 "available" for ADN located under the local phonebook;
- Presence of EF_{ADN} in EF_{PBR} for ADN located under the global phonebook;
- Presence of EF_{ANR} in EF_{PBR} for ANR;
- Service n2 "available" for FDN;
- Service n4 "available" for SDN;
- Service n27 "available" for EF_{OCI};
- Service n28 "available" for EF_{ICI}.

The following procedures may not only be applied to EF_{ADN} and its associated extension files EF_{CCP1} and EF_{EXT1} as described in the procedures below, but also to EF_{ANR}, EF_{FDN}, EF_{SDN}, EF_{OCI}, and EF_{ICI}, and their associated extension files. If these files are not available, as denoted in the CSIM 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.

Update: The ME analyzes and assembles the information to be stored as follows (the byte identifiers used below corresponds to those in the definition of the relevant EFs in the present document):

i) The ME identifies the Alpha-tagging, Capability/Configuration1 Record Identifier and Extension1 Record Identifier.

ii) The dialing number/SSC string shall be analyzed and allocated to the bytes of the EF as follows:

- if a "+" is found, the TON identifier is set to "International";
- if 20 or less "digits" remain, they shall form the dialing number/SSC string;
- if more than 20 "digits" remain, the procedure shall be as follows:
- 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 first 20 "digits" are stored in the dialing 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 identifier is coded with the associated record number in the EF_{EXT1}. The remaining digits are stored in the selected Extension1 record where the type of the record is set to "additional data". The first byte of the Extension1 record is set with the number of bytes of the remaining additional data. The number of bytes containing digit information is the sum of the length of BCD number/SSC contents of EF_{ADN} and byte 2 of all associated chained Extension1 records containing additional 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:
- 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:
  - The ME seeks for two free records in EF_{EXT1}. 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 EF_{EXT1} 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 EF_{ADN}. If the CSIM has no available empty space to store the received ADN/SSC, or if the procedure has been aborted, the ME advises the user.

For reasons of memory efficiency, the ME may analyze all Extension1 records to recognize if the additional or subaddress data to be stored already exists in EF_{EXT1}. In this case, the ME may use the existing chain or the last part of the existing chain from more than one ADN. The ME is only allowed to store extension data in unused records. If existing records are used for multiple accesses, the ME shall not change any data in those records to prevent corruption of existing chains.

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".

Request: The ME sends the identification of the information to be read. The ME shall analyze the data of EF_{ADN} to ascertain~~,~~ whether additional data is associated in EF_{EXT1} or EF_{CCP1}. If necessary, then the ME performs the reading procedure on these EFs to assemble the complete ADN/SSC.

Purge: The ME shall access each EF which references EF_{EXT1} 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 records are noted by the ME. All Extension1 records not noted are then marked by the ME as "free" by setting the whole record to 'FF'.

The following three procedures are only applicable to service n2 (FDN).

FDN capability request. If the FDN service is available (see section 5.2.18), the ~~The~~ ME shall check the ~~state of service n2, i.e.~~ if the FDN service is "enabled" or "disabled" (see section 5.2.80 EF_{EST}). If the FDN service is "enabled", the ME shall only allow outgoing calls to one of the numbers in EF_{FDN}. ~~To ascertain the state of FDN, the ME shall check in EF_{CSIM_ST} and EF_{EST} if FDN is enabled (service "activated" and "available"). In all other cases service n2 is "disabled".~~

~~FDN enabling is done by activating the FDN service in EF_{EST}.~~

~~FDN disabling is done by deactivating the FDN service in EF_{EST}.~~

7.3.3 Short Message

Requirement: Service n6 "available".

Request: The CSIM seeks for the identified short message. If this message is found, the ME performs the reading procedure with EF_{SMS}.

If the short message is not found within the CSIM memory, the CSIM indicates that to the ME.

Update: The ME looks for the next available area to store the short message. If such an area is available, it performs the updating procedure with EF_{SMS}.

If there is no available empty space in the CSIM to store the received short message, a specific ~~MMI~~ user interaction will have to take place in order not to lo~~o~~se the message.

Erasure: The ME will select in the CSIM the message area to be erased. Depending on the ~~MMI~~UI, the message may be read before the area is marked as "free". After performing the updating procedure with EF_{SMS}, the memory allocated to this short message in the CSIM is made available for a new incoming message. The memory of the CSIM may still contain the old message until a new message is stored in this area.

If b6 of byte 1 in EF_{SMS} is set to '1' (the message in the corresponding record is protected), then a specific ~~MMI~~ user interaction may take place in order not to lose the message.

7.3.4 ~~Capability configuration parameters~~Void

~~Requirement: Service n33 "available".~~

~~Request: The ME performs the reading procedure with EF_{CCP2}.~~

~~Update: The ME performs the updating procedure with EF_{CCP2}.~~

~~Erasure: The ME sends the identification of the requested information to be erased. The content of the identified record in EF_{CCP2} is marked as "free".~~

7.3.5 Group Identifier level 1

Requirement: Service n23 "available".

Request: The ME performs the reading procedure with EF_{GID1}.

7.3.6 Group Identifier level 2

Requirement: Service n24 "available".

Request: The ME performs the reading procedure with EF_{GID2}.

7.3.7 Service provider name

Requirement: Service n10 "available".

Request: The ME performs the reading procedure with EF_{SPN}.

7.3.8 Depersonalisation Control Keys

Requirement: Service n25 "available".

Request: The ME performs the reading procedure with EF_{DCK}.

7.3.9 Co-operative Network List

Requirement: Service n26 "available".

Request: The ME performs the reading procedure with $EF_{CDMACNL}$.

~~### 7.3.10 Enabled Services Table Request~~

~~Requirement: Service n32 "available".~~

~~Request: The ME performs the reading procedure with EF_{EST}.~~

~~Update: The ME performs the updating procedure with EF_{EST}.~~

~~7.3.11~~7.3.10 MMS Notifications

Requirement: Service n19 "available".

Request: The ME sends the identification of the information to be read, and then the ME performs the reading procedure with EF_{MMSN}. If Service n20 is available the ME shall analyze the data of EF_{MMSN} to ascertain, whether additional data is associated in EF_{EXT8}. If necessary, then the ME performs the reading procedure on EF_{EXT8} to assemble the complete MMS notification.

Update: The ME analyzes and assembles the MMS notification to be stored as follows:

- if the MMS notification contains not more bytes than the maximum possible number for EF_{MMSN} then the ME looks for the next available area to store the MMS notification. If such an area is available, it performs the updating procedure with EF_{MMSN}.
- if the MMS notification contains more bytes than the maximum possible number for EF_{MMSN} then the ME seeks for a sufficient number of free records in EF_{EXT8} to store the complete MMS notification.
  - If there is not a sufficient number of EF_{EXT8} records marked as "free" to store the complete MMS notification, the procedure is aborted.
  - Otherwise, the ME performs the updating procedure and stores as many bytes as possible in EF_{MMSN}. The Extension file record number of EF_{MMSN} is coded with the associated record number in the EF_{EXT8}. The remaining bytes are stored in the selected EF_{EXT8} record where the type of the record is then set to "additional data". The second byte of the EF_{EXT8} record is set with the number of bytes of the remaining additional data. It is possible, if the number of additional digits exceeds the capacity of the additional record, to chain another record inside the EF_{EXT8} by the identifier in the last byte of the record. In this case byte 2 of each record for additional data within the same chain indicates the number of bytes within the same record.

    The ME is only allowed to store extension data in unused records of EF_{EXT8}

If there is no available empty space in the CSIM to store the MMS notification, it is up to ME implementation how the notification is handled.

Erasure: The ME will select in the CSIM the MMS notification to be erased. Depending on the ~~MMI~~UI, the MMS notification may be read before the area is marked as "free". The memory of the CSIM may still contain the old MMS notification until a new message is stored. If Service n20 is available all associated records in EF_{EXT8} are then marked by the ME as "free" by setting them to 'FF'.

~~7.3.12~~7.3.11 MMS Issuer Connectivity Parameters

Requirement: Service n19 "available".

Request: the ME performs the reading procedure with EF_{MMSICP}.

Update: The ME performs the updating procedure with EF_{MMSICP}.

~~7.3.13~~7.3.12 MMS User Preferences

Requirement: Service n19 "available".

Request: the ME performs the reading procedure with EF_{MMSUP}.

Update: The ME performs the updating procedure with EF_{MMSUP}.

~~7.3.14~~7.3.13 MMS User Connectivity Parameters

Requirement: Service n19 and n21 "available".

Request: the ME performs the reading procedure with EF_{MMSUCP}.

Update: The ME performs the updating procedure with EF_{MMSUCP}.

~~7.3.15~~7.3.14 Multimedia Message Storage

If the ME supports Multimedia Message Storage on the CSIM, then the following procedures apply. As defined in [37] a Multimedia Message (MM) consists of content, or multimedia objects, and headers to describe various properties of that content. An MM is stored in EF_{MMDF}[30], a BER-TLV structured file.

A list of multimedia messages is stored in the BER-TLV file EF_{MML} [30] where each data object identifies one Multimedia Message stored in EF_{MMDF}[30].

Requirement: Service n30 "available".

Request: The ME performs the reading procedures on EF_{MML}[30] to verify the presence and to get the location information of the targeted MM. Then the ME performs the reading procedure of the EF_{MMDF}[30] file to get the MM.

Update: The ME chooses a free identity (i.e. not listed in EF_{MML}[30]) for the multimedia message and check for available space in the EF_{MMDF}[30] file. This procedure could be done for each update or once at the startup of the UE and after a REFRESH command involving one of the $DF_{MULTIMEDIA}$ files. Then the ME performs the following procedures:

If there is no available empty space in the EF_{MMDF}[30] file to store the MM, the procedure is aborted and the user is notified.

Else, the ME stores the MM in EF_{MMDF}[30], then updates the information in EF_{MML}[30] accordingly.

Erasure: After a successful deletion of an MM in EF_{MMDF}[30] the ME updates the information in EF_{MML}[30] accordingly.

7.4 CCAT Related Procedures

7.4.1 Data Download via SMS-PP

Requirement: Service n12 "available".

Procedures and commands for Data Download via SMS-PP are defined in [47].

7.4.2 Data Download via SMS Broadcast

Requirement: Service n11 "available".

Procedures and commands for Data Download via SMS Broadcast are defined in [47].

7.4.3 Call Control by CSIM

Requirement: ~~Service n13 "available".~~

For Voice Call Control, Service n13 “available”.

For Data Call Control, Service n42 “available”.

For Mobile Originated SMS Call Control, Service n43 “available”.

Procedures and commands for Call Control by CSIM are defined in [47].

7.4.4 Image Request

The ME sends the identification of the information to be read. The ME shall analyze the data of EF_{IMG} to identify the files containing the instances of the image. If necessary, then the ME performs READ BINARY commands on these files to assemble the complete image instance data.

8. ~~STRUCTURE~~ STRUCTURE OF ~~COMMANDS~~ COMMANDS AND ~~RESPONSES~~RESPONSES

This section defines the command and response APDU's supported by the UICC.

8.1 Command APDU Structure

See [18] section 10.1

8.1.1 Coding of Class byte

See [18] Section 10.1.1 for telecom application and Table 1, below, for CSIM commands.

8.1.2 Coding of Instruction byte

~~8.1.3~~8.1.2.1 Coding of Instruction byte for a telecom application.

See [18] Section 10.1.2

~~8.1.4~~8.1.2.2 Coding of Instruction byte for CSIM

Table 1 depicts coding of additional instruction byte of the commands for CSIM.

Table 1 Coding of ~~additional Instruction Byte~~CLA, INS, P1 and P2 Bytes of the CSIM Commands ~~for a CSIM~~

~~COMMAND~~	~~CLA~~	~~INS~~
~~Command APDUs~~		
~~Security-related commands~~		
~~Manage SSD (Update & Confirm SSD)~~	~~8X~~	~~'82'~~
~~Base Station Challenge~~	~~8X~~	~~'8A'~~
~~Generate Key / VPM~~	~~8X~~	~~'8E'~~
~~Authenticate~~	~~0X~~	~~'88'~~
~~OTASP/OTAPA-related commands~~		
~~Generic Key Generation Request~~	~~8X~~	~~'50'~~
~~Commit~~	~~8X~~	~~'CC'~~
~~Validate~~	~~8X~~	~~'CE'~~
~~Generic Configuration Request~~	~~8X~~	~~'54'~~
~~Generic Download Request~~	~~8X~~	~~'56'~~

~~COMMAND~~	~~CLA~~	~~INS~~
~~Command APDUs~~		
~~OTAPA Request~~	~~8X~~	~~'EE'~~
~~Secure Mode~~	~~8X~~	~~'4A'~~
~~FRESH~~	~~8X~~	~~'4C'~~
~~**ESN Management command**~~		
~~Store ESN_MEID_ME~~	~~8X~~	~~'DE'~~
~~**Packet Data Security-related command**~~		
~~Compute IP Authentication~~	~~8X~~	~~'80'~~
~~**BCMCS-related command**~~		
~~BCMCS~~	~~8X~~	~~'58'~~
~~**Application Authentication command**~~		
~~Application Authentication~~	~~8X~~	~~'5A'~~
~~**AKA-related commands**~~		
~~UMAC Generation~~	~~8X~~	~~'5E'~~
~~CONFIRM_KEYS~~	~~8X~~	~~'5C'~~
~~**LCS-related commands**~~		
~~S-SAFE Verification & Decryption~~	~~8X~~	~~'40'~~
~~TLS Generate Master Secret~~	~~8X~~	~~'42'~~
~~TLS Generate Verify_data~~	~~8X~~	~~'44'~~
~~TLS Verification and Generate key_block~~	~~8X~~	~~'46'~~

1

COMMAND	CLA	INS	P1	P2
Command APDUs				
Security-related commands				
MANAGE SSD				
UPDATE SSD	8X	'82'	'00'	'00'
CONFIRM SSD			'00'	'01'
BASE STATION CHALLENGE	8X	'8A'		
GENERATE KEY / VPM	8X	'8E'		

COMMAND	CLA	INS	P1	P2
Command APDUs				
AUTHENTICATE	0X	'88'	'00'	
RUN CAVE				'80'
3G ACCESS AKA				'81'
EAP AKA				'82'
OTASP/OTAPA-related commands				
GENERIC KEY GENERATION REQUEST	8X	'50'	'00'	
MS KEY REQUEST				'00'
KEY GENERATION REQUEST				'01'
SERVICE KEY GENERATION REQUEST				'02'
COMMIT	8X	'CC'		
VALIDATE	8X	'CE'		
GENERIC CONFIGURATION REQUEST	8X	'54'	'00'	
CONFIGURATION REQUEST				'00'
SSPR CONFIGURATION REQUEST				'01'
PUZL CONFIGURATION REQUEST				'02'
3GPD CONFIGURATION REQUEST				'03'
MMS CONFIGURATION REQUEST				'04'
SYSTEM TAG CONFIGURATION REQUEST				'05'
MMSS CONFIGURATION REQUEST				'06'
GENERIC DOWNLOAD REQUEST	8X	'56'	'00'	
DOWNLOAD REQUEST				'00'
SSPR DOWNLOAD REQUEST				'01'
PUZL DOWNLOAD REQUEST				'02'
3GPD DOWNLOAD REQUEST				'03'
MMS DOWNLOAD REQUEST				'04'
SYSTEM TAG DOWNLOAD REQUEST				'05'
MMSS DOWNLOAD REQUEST				'06'

COMMAND	CLA	INS	P1	P2
Command APDUs				
OTAPA REQUEST	8X	'EE'	'00'	'00'
SECURE MODE	8X	'4A'		
FRESH	8X	'4C'		
ESN Management command				
STORE ESN_MEID_ME	8X	'DE'		
Packet Data Security-related command				
COMPUTE IP AUTHENTICATION	8X	'80'		
BCMCS-related command				
BCMCS	8X	'58'		
Application Authentication command				
APPLICATION AUTHENTICATION	8X	'5A'		
AKA-related commands	8X			
UMAC GENERATION		'5E'		
CONFIRM_KEYS		'5C'		
LCS-related commands	8X			
S-SAFE VERIFICATION AND DECRYPTION		'40'	'00'	'00'
TLS GENERATE MASTER SECRET		'42'		
TLS GENERATE VERIFY DATA		'44'		'00'
TLS VERIFICATION AND GENERATE KEY BLOCK		'46'		'00'

~~8.1.5~~8.1.3 Coding of Parameter bytes

The value of the parameters P1 and P2 depends on the command. If these are a constant value they are specified in Table 1, above, otherwise in section 9 below.~~If the parameter is not used, the value is set to '00'. Coding of the parameter bytes is presented in Section 8.~~

~~8.1.6~~8.1.4 Coding of Lc bytes

This field encodes the length of the command parameters/data. See [18] Section 10.1.4 and section 9, below.

~~8.1.7~~8.1.5 Coding of Data part

See [18] Section 10.1.5 and section 9, below.

~~8.1.8~~8.1.6 Coding of Le bytes

This field encodes the maximum length of the command parameters/data. See [18] Section 10.1.6 and section 9, below.

8.2 Response APDU structure

See [18] Section 10.2 and section 9, below.

9. ~~COMMANDS~~COMMANDS

9.1 Generic Commands

See [18] Section 11.1.

9.2 CAT Commands

See [18] Section 11.2.

9.3 Data Oriented Commands

See [18] Section 11.3.

9.4 CSIM Commands

This section describes the APDU commands, which ~~is~~ are only applicable for CSIM. These commands are related to a particular CSIM and shall not be executable unless the CSIM application has been selected and activated, and the current directory is the CSIM ADF or any subdirectory under this ADF and a successful PIN verification procedure has been performed (see Section 7).

9.4.1 Security-related Commands

The commands BASE STATION CHALLENGE, UPDATE SSD and CONFIRM SSD ~~*Base Station Challenge, Update SSD* and *Confirm SSD*~~ are performed in sequence, as described in [46] Section 4.2, 4.4 and ~~4.4~~ Annex D.

If the CSIM receives a BASE STATION CHALLENGE command, it shall re-start the command sequence.

If T=0 protocol is used, APDU is mapped onto TPDU (see Section 7.3.1.1 in [18]).

~~9.4.2~~9.4.1.1 ~~Manage~~ MANAGE SSD

~~9.4.2.1.1~~9.4.1.1.1 Functional Description

MANAGE SSD consists of UPDATE SSD and CONFIRM SSD commands (see [46] Section 4.2).~~Manage SSD consists of *Update SSD* and *Confirm SSD* command (see [46] Section 4.2).~~

~~They are differentiated by P2 value (see Section 9.4.1.1.2).~~

9.4.2.1.2 Command parameters and data

Code	Value
CLA	As specified in Section 8.1.1
INS	As specified in Section 8.1.2
P1	'00'
P2	See Table 2
Lc	Length of the subsequent data field
Data	*Update SSD* or *Confirm SSD* related data
Le	Not present for both *Update SSD* and *Confirm SSD* command

Table 2 Coding of P2 of Manage SSD Command

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	0	*Update SSD* command
0	0	0	0	0	0	0	1	*Confirm SSD* command

a. UPDATE SSD *Update SSD* command data (P2='00')

Coded as [46] Section 4.4.1 (UPDATE SSD) except that the coding of CLA, INS, P1 and P2 is defined in Table 1. The command parameters/data and response parameters/data are coded as [46] Section 4.4.1 (Update SSD).

b. CONFIRM SSD *Confirm SSD* command data (P2='01')

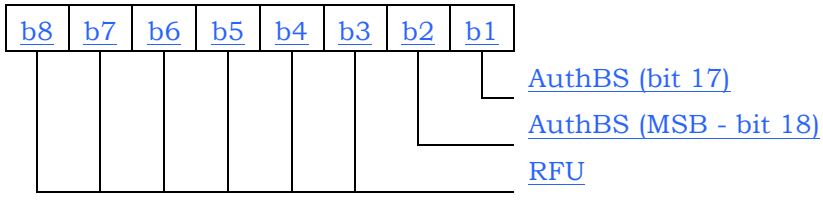
COMMAND	CLASS	INS	P1	P2	Lc	Le
CONFIRM SSD	'8X'	'82'	'00'	'00'	'03'	absent

Command parameters/data:

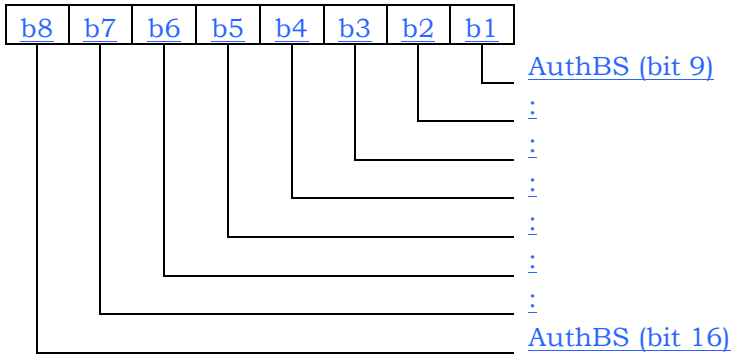
Octet(s)	Description	Length
1 - 3	AuthBS	3 bytes

AuthBS shall be coded as follows:

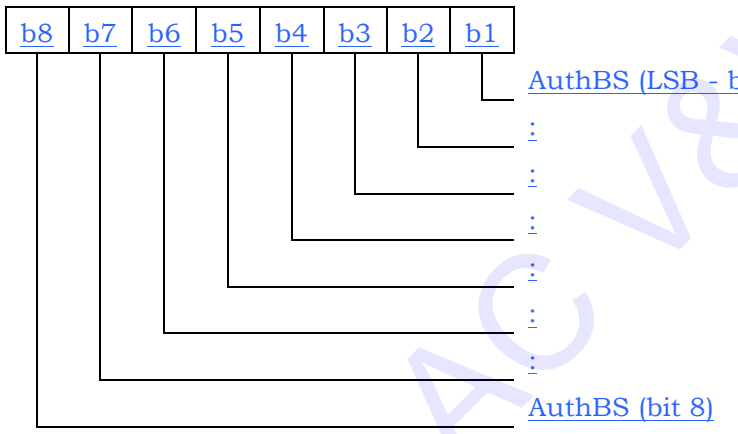
Octet 1:



Octet 2:



Octet 3:



Response parameters/data:

No response parameters are generated as a result of command execution. Successful comparison will cause SW1/SW2 to be set to '90'/'00'. Unsuccessful comparison will cause SW1/SW2 to be set to '69'/'82' (Error - Security status not satisfied) [18] Table 10.13).

If the ME is assigned an MEID and if bit1 of the EF_{USGIN} is set to '0', then the pESN value received in the UPDATE SSD command shall be used as the ESN input to the CAVE algorithm for the computation of AuthBS.

~~The command parameters/data and response parameters/data are coded as [46] Section 4.4.3 (Confirm SSD).~~

9.4.1.2 ~~Base Station Challenge~~BASE STATION CHALLENGE

~~9.4.2.1.3~~9.4.1.2.1 Functional Description

The function of the BASE STATION CHALLENGE ~~Base Station Challenge~~ command is described in [46] Section 4.2.1 and 4.4.

~~9.4.2.1.4~~9.4.1.2.2 Command parameters and data

Coded as [46] Section 4.4.2, except that CLA and INS bytes shall follow Table 1.

~~The command parameters/data and response parameters/data are coded as [46] Section 4.4.2, where CLA and INS byte shall follow Section 8.1.1, and Le is the length of data expected in response (= '04').~~

9.4.1.3 GENERATE KEY/VPM~~Generate Key/VPM~~

~~9.4.2.1.5~~9.4.1.3.1 Functional Description

The function of the GENERATE KEY/VPM ~~*Generate Key/VPM*~~ command is described in [46] Section 4.2.2.

This command relies on the prior successful execution of the AUTHENTICATE - RUN CAVE command with the "save" function activated (bit 4 of the *Process_Control* parameter). If this has not occurred, the status word SW1='98' and SW2='34' [see section 4.2] shall be returned upon the invocation of this command.

~~This command relies on the prior successful execution of the *Authenticate - Run CAVE* command with the "save" function activated (bit 4 of *Process_Control* parameter). If this has not occurred, the status word SW='98' and SW='34' shall be returned upon the invocation of this command.~~

~~9.4.2.1.6~~9.4.1.3.2 Command parameters and data

Coded as [46] Section 4.4.5, except that CLA and INS byte shall follow Table 1.

~~The command parameters/data and response parameters/data are coded as [46] Section 4.4.5, where CLA and INS byte shall follow Section 8.1.1, and Le is '00' or maximum the length of data expected in response.~~

~~9.4.2.2~~9.4.1.4 AUTHENTICATE~~Authenticate~~

~~9.4.2.2.1~~9.4.1.4.1 Functional Description

This command performs several authentication functions, i.e.: RUN CAVE, 3G ACCESS AKA, and EAP AKA (see [46] Section 4.4.4).~~This command performs several authentication functions, i.e.: *Run CAVE, 3G Authentication AKA*, and *WLAN Authentication AKA*(see [46] Section 4.4.4.)~~

~~They are differentiated by P2 value (see Section 9.4.1.4.2).~~

~~9.4.2.2.2 Command parameters and data~~

~~Code~~	~~Value~~
~~CLA~~	~~As specified in Section 8.1.1~~
~~INS~~	~~'88'~~
~~P1~~	~~'00'~~
~~P2~~	~~See Table 3~~
~~Le~~	~~See below~~
~~Data~~	~~See below~~
~~Le~~	~~'00', or maximum length of data expected in response~~

Table 3 Coding of P2 of Authenticate Command

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
1	-	-	-	-	-	-	-	Specific reference data (e.g. DF specific/application dependant key)
1	0	0	0	0	0	0	0	*Run CAVE*
1	0	0	0	0	0	0	1	*3G Authentication AKA*
1	0	0	0	0	0	1	0	*WLAN Authentication AKA*

a. *Run* RUN CAVE command data (P2='80')

Coded as [46] Section 4.4.4 "Run CAVE" except that CLA, INS, P1 and P2 are defined in Table 1. The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.4.4

b. 3G ACCESS AKA *3G Authentication AKA* command data (P2='81')

Coded as [46] Section 4.4.4 "3G Access AKA" except that CLA, INS, P1 and P2 are defined in Table 1. The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.4.4

c. EAP AKA *WLAN Authentication AKA* command data (P2='82)

Coded as [46] Section 4.4.4 "EAP AKA" except that CLA, INS, P1 and P2 are defined in Table 1. The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.4.4

9.4.3 9.4.2 OTASP/OTAPA-related Commands

This section specifies the CSIM commands which are the mapping of "Request/Response" messages described in [7] and [46] Section 4.3.

9.4.2.1 GENERIC KEY GENERATION REQUEST Generic Key Generation

9.4.3.1.1 9.4.2.1.1 Functional Description

This command performs several key generation functions, i.e.: MS KEY REQUEST, KEY GENERATION REQUEST, and SERVICE KEY GENERATION REQUEST, which correspond to the *MS Key Request/Response*, *Key Generation Request/Response* and *Service Key Generation Request/Response* messages specified in [7].

This command performs several key generation functions, i.e.: *MS Key Request*, *Key Generation Request*, and *Service Key Generation Request*, which corresponds to *MS Key Request/Response*, *Key Generation Request/Response* and *Service Key Generation Request/Response* messages specified in [7].

Those key generation functions are differentiated by P2 value (see Section 9.4.2.1.2).

As specified in [7], the KEY GENERATION REQUEST follows the MS KEY REQUEST function.~~As specified in [7], *MS Key Request* function relates to *Key Generation Request* function in a way that *Key Generation Request* follows the *MS Key Request* function.~~

~~9.4.3.1.2 Command parameters and data~~

~~Code~~	~~Value~~
~~CLA~~	~~As specified in Section 8.1.1~~
~~INS~~	~~'50'~~
~~P1~~	~~'00'~~
~~P2~~	~~See Table 4~~
~~Lc~~	~~See below~~
~~Data~~	~~See below~~
~~Le~~	~~'00', or maximum length of data expected in response~~

~~**Table 4 Coding of P2 of Generic Key Command**~~

~~b8~~	~~b7~~	~~b6~~	~~b5~~	~~b4~~	~~b3~~	~~b2~~	~~b1~~	~~Meaning~~
~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~MS Key Request~~
~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~Key Generation Request~~
~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~0~~	~~Service Key Generation Request~~

a. ~~MS Key Request~~MS KEY REQUEST ~~command data (P2='00')~~

Coded as [46] Section 4.5.1 except that CLA, INS, P1 and P2 are defined in Table 1.~~The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.1~~

b. ~~Key Generation Request command data (P2='01')~~KEY GENERATION REQUEST

Coded as [46] Section 4.5.2 except that CLA, INS, P1 and P2 are defined in Table 1.~~The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.2~~

c. ~~*Service Key Generation Request* command data (P2='02')~~SERVICE KEY GENERATION REQUEST

Coded as [46] Section 4.5.16 except that CLA, INS, P1 and P2 are defined in Table 1.~~The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.16~~

9.4.2.2 ~~Commit~~COMMIT

~~9.4.3.1.3~~9.4.2.2.1 Functional Description

This command corresponds to the *Commit Request/Response* messages specified in [7], Sections 4.5.1.6 and 3.5.1.6, respectively.

~~9.4.3.1.4~~9.4.2.2.2 Command parameters and data

Coded as [46] Section 4.5.3, except that CLA and INS are defined in Table 1.~~The response parameters/data are coded as [46] Section 4.5.3, where CLA and INS byte shall follow Section 8.1.1, Lc is not present, and Le is length of expected data in response (= '01').~~

~~9.4.3.2~~9.4.2.3 ~~Validate~~VALIDATE

~~9.4.3.2.1~~9.4.2.3.1 Functional Description

This command requests ~~a~~ the validation of a single block of data and forms a subset of the *Validation Request Message* as described in [7], Section 4.5.1.10. ~~And the~~The response pertains to a single block of data and forms a subset of the *Validation Response Message* as described in [7], Section 3.5.1.10.

~~9.4.3.2.2~~9.4.2.3.2 Command parameters and data

Coded as [46] Section 4.5.4, except that the CLA and INS bytes are defined in Table 1.~~The command parameters/data and response parameters/data are coded as [46] Section 4.5.4, where CLA and INS byte shall follow Section 8.1.1, and Le is length of the data expected in response (= '02').~~

~~9.4.3.3~~9.4.2.4 GENERIC CONFIGURATION REQUEST~~Generic Configuration Request~~

~~9.4.3.3.1~~9.4.2.4.1 Functional Description

This command performs several 'configuration request' functions: CONFIGURATION REQUEST, SSPR CONFIGURATION REQUEST, PUZL CONFIGURATION REQUEST, 3GPD CONFIGURATION REQUEST, MMS CONFIGURATION REQUEST, SYSTEM TAG CONFIGURATION REQUEST and MMSS CONFIGURATION REQUEST which correspond to *Configuration Request/Response, SSPR Configuration Request/Response, PUZL Configuration Request/Response, 3GPD Configuration Request/Response* messages, *MMS Configuration Request/ Response, System Tag Configuration Request/ Response* and MMSS Configuration Request/Response as specified in [7].~~This command performs several 'configuration request' functions, i.e.: *Configuration Request, SSPR Configuration Request, PUZL Configuration Request, 3GPD Configuration Request, MMS Configuration Request* and *System Tag Configuration Request* which corresponds to *Configuration Request/Response, SSPR Configuration Request/Response, PUZL Configuration Request/Response, 3GPD Configuration Request/Response* messages, *MMS Configuration Request/ Response, System Tag Configuration Request/ Response* and MMSS Configuration Request/Response as specified in [7].~~

~~Those 'configuration request' functions are differentiated by P2 value (see Section 9.4.2.4.2).~~

9.4.3.3.2 Command parameters and data

Code	Value
CLA	As specified in Section 8.1.1
INS	'54'
P1	'00'
P2	See Table 5
Lc	See below
Data	See below
Le	'00', or maximum length of data expected in response

Table 5 Coding of P2 of Generic Configuration Request

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	0	Configuration Request
0	0	0	0	0	0	0	1	SSPR Configuration Request
0	0	0	0	0	0	1	0	PUZL Configuration Request
0	0	0	0	0	0	1	1	3GPD Configuration Request
0	0	0	0	0	1	0	0	MMS Configuration Request
0	0	0	0	0	1	0	1	System Tag Configuration Request
0	0	0	0	0	1	1	0	MMSS Configuration Request

a. CONFIGURATION REQUEST *Configuration Request* command data (P2='00')

Coded as [46] Section 4.5.5 except that CLA, INS, P1 and P2 are defined in Table 1. The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.5.

b. SSPR CONFIGURATION REQUEST SSPR Configuration Request command data (P2='01')

Coded as [46] Section 4.5.7 except that CLA, INS, P1 and P2 are defined in Table 1. The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.7.

c. PUZL CONFIGURATION REQUEST PUZL Configuration Request command data (P2='02')

Codedas [46] Section 4.5.10 except that CLA, INS, P1 and P2 are defined in Table 1. The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.10.

d. 3GPD CONFIGURATION REQUEST 3GPD Configuration Request command data (P2='03')

Codedas [46] Section 4.5.12 except that CLA, INS, P1 and P2 are defined in Table 1. The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.12.

e. MMS CONFIGURATION REQUEST~~MMS Configuration Request command data (P2='04')~~

Codedas [46] Section 4.5.19 except that CLA, INS, P1 and P2 are defined in Table 1.~~The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.19.~~

f. SYSTEM TAG CONFIGURATION REQUEST~~System Tag Configuration Request command data (P2='05')~~

Codedas [46] Section 4.5.21 except that CLA, INS, P1 and P2 are defined in Table 1.~~The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.21.~~

g. MMSS CONFIGURATION REQUEST~~MMSS Configuration Request command data (P2='06')~~

CLA, INS, P1 and P2 are defined in Table 1. Lc is the length of the command parameters/data (4) and Le is the maximum length of the response parameters/data (Nr or greater).

The command parameters/data, input parameters and response parameters/data are coded as specified below.

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2 – 3	Request Offset	2 bytes
4	Request Max Size	1 byte

This command requests MMSS configuration details of a single block of data and forms a subset of the "MMSS Configuration Request Message" as described in [7], section 4.5.1.25.

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte
3	Block Length	1 byte
4 – ~~Le~~Nr	Param Data	~~Le~~ Nr – 3 bytes

~~* Note: Le = Length of Param Data + 3.~~

This response provides MMSS configuration details of a single block of data and forms a subset of the "MMSS Configuration Response Message" as described in [7], section 3.5.1.25.

9.4.2.5 GENERIC DOWNLOAD REQUEST~~Generic Download Request~~

~~9.4.3.3.3~~9.4.2.5.1 Functional Description

This command performs several 'download request' functions: DOWNLOAD REQUEST, SSPR DOWNLOAD REQUEST, PUZL DOWNLOAD REQUEST, 3GPD DOWNLOAD REQUEST, MMS DOWNLOAD REQUEST , System Tag DOWNLOAD REQUEST Request and MMSS DOWNLOAD REQUEST which correspond to *Download Request/Response*, *SSPR Download Request/Response*, *PUZL Download Request/Response* and *3GPD Download Request/Response* messages, *MMS Download Request/ Response*, *System Tag Download Request/ Response* and *MMSS Download Request/Response* as specified in [7].~~This command performs several 'download request' functions, i.e.: *Download Request*, *SSPR Download Request*, *PUZL Download Request*, *3GPD Download Request*, *MMS Download Request* and *System Tag Download Request* which corresponds to *Download Request/Response*, *SSPR Download Request/Response*, *PUZL Download Request/Response* and *3GPD Configuration Request/Response* messages, *MMS Configuration Request/ Response*, *System Tag Configuration Request/ Response* and MMSS Download Request/Response as specified in [7].~~

~~Those 'download request' functions are differentiated by P2 value (see Section 9.4.2.5.2).~~

~~9.4.3.3.4 Command parameters and data~~

~~Code~~	~~Value~~
~~CLA~~	~~As specified in Section 8.1.1~~
~~INS~~	~~'56'~~
~~P1~~	~~'00'~~
~~P2~~	~~See Table 6~~
~~Lc~~	~~See below~~
~~Data~~	~~See below~~
~~Le~~	~~Maximum length of data expected in response~~

~~Table 6 Coding of P2 of Generic Download Request~~

~~b8~~	~~b7~~	~~b6~~	~~b5~~	~~b4~~	~~b3~~	~~b2~~	~~b1~~	~~Meaning~~
~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~Download Request~~
~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~SSPR Download Request~~
~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~0~~	~~PUZL Download Request~~
~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~1~~	~~3GPD Download Request~~
~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~0~~	~~0~~	~~MMS Download Request~~
~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~0~~	~~1~~	~~System Tag Download Request~~
~~0~~	~~0~~	~~0~~	~~0~~	~~0~~	~~1~~	~~1~~	~~0~~	~~MMSS Download Request~~

a. DOWNLOAD REQUEST ~~*Download Request* command data (P2='00')~~

Coded as [46] Section 4.5.6 except that CLA, INS, P1 and P2 are defined in Table 1. ~~The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.6~~

b. SSPR DOWNLOAD REQUEST ~~*SSPR Download Request* command data (P2='01')~~

Coded as [46] Section 4.5.8 except that CLA, INS, P1 and P2 are defined in Table 1. ~~The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.8~~

c. PUZL DOWNLOAD REQUEST ~~*PUZL Download Request* command data (P2='02')~~

Coded as [46] Section 4.5.11 except that CLA, INS, P1 and P2 are defined in Table 1. ~~The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.11~~

d. 3GPD DOWNLOAD REQUEST ~~*3GPD Download Request* command data (P2='03')~~

Coded as [46] Section 4.5.13 except that CLA, INS, P1 and P2 are defined in Table 1. ~~The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.13~~

e. MMS DOWNLOAD REQUEST ~~MMS Download Request command data (P2='04')~~

Coded as [46] Section 4.5.20 except that CLA, INS, P1 and P2 are defined in Table 1. ~~The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.20~~

f. SYSTEM TAG DOWNLOAD REQUEST ~~System Tag Download Request command data (P2='05')~~

Coded as [46] Section 4.5.21 except that CLA, INS, P1 and P2 are defined in Table 1. ~~The command parameters/data, input parameters and response parameters/data are coded as [46] Section 4.5.22~~

g. MMSS DOWNLOAD REQUEST ~~MMSS Download Request command data (P2='06')~~

CLA, INS, P1 and P2 are defined in Table 1. Lc is the length of the command parameters/data (4) and Le is the maximum length of the response parameters/data (Nr or greater).

The command parameters/data, input parameters and response parameters/data are coded as below:

Command parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Block Length	1 byte
3 – ~~Le~~Nr	Param Data	~~Le~~ Nr – 2 bytes

This command requests the MMSS download of a single block of data and forms a subset of the "MMSS Download Request Message" as described in [7], section 4.5.1.26.

~~* Note: Lc = Length of Param Data + 2.~~

Response parameters/data:

Octet(s)	Description	Length
1	Block ID	1 byte
2	Result Code	1 byte
3 – 4	Segment Offset	2 bytes
5	Segment Size	1 byte

Details of the response are in [7], section 3.5.1.9, "MMSS Download Response Message".

9.4.2.6 OTAPA REQUESTOTAPA Request

9.4.3.3.59.4.2.6.1 Functional Description

This command corresponds to the *OTAPA Request/Response* messages specified in [7], Sections 4.5.1.11 and 3.5.1.11, respectively.

9.4.3.3.69.4.2.6.2 Command parameters and data

CLA, INS, P1 and P2 are defined in Table 1. Lc is the length of the command parameters/data (12) and Le is the maximum length of the response parameters/data.The command parameters/data and response parameters/data are coded as mentioned below, where CLA and INS byte shall follow Section 8.1.1, and Le is the length of the data expected in response (= '06').

Code	Value
CLA	As specified in Section 8.1.1
INS	'EE'
P1	'00'
P2	'00'
Le	'0C'
Data	See below
Le	'06'

Command parameters/data:

Octet(s)	Description	Length
1	Start/Stop	1 byte
2 – 5	RANDSeed	4 bytes
6-129	ESN/Pseudo pESN	7 bytes
10-12	RFU	3 bytes

The Start/Stop parameter as defined in Section 4.5.1.11 of [7] shall be coded as follows:

Octet 1

8	7	6	5	4	3	2	1

'0'RFU
'0'
'0'
'0'
'0'
'0'
'0'
Start/Stop field

Response parameters/data:

Octet(s)	Description	Length
1	Result Code	1 byte
2	NAM_LOCK Indicator	1 byte
3 – 6	RAND OTAPA	4 bytes

The RAND_OTAPA (bytes 3-6) is returned if and only if the Result_Code is '00' and the NAM_LOCK_STATE is enabled (='1') and the Start/Stop field was set to '1' (Start) in the OTAPA REQUEST command.

The NAM_LOCK Indicator parameter as defined in Section 3.5.1.11 of [7] shall be coded as follows:

Octet 2

8	7	6	5	4	3	2	1

NAM_LOCK Indicator
~~'0'~~RFU
~~'0'~~
~~'0'~~
~~'0'~~
~~'0'~~
~~'0'~~
~~'0'~~

Details of the response are in [7], sections 3.3.1.11, "OTAPA Request Message Processing", and 3.5.1.11, "OTAPA Response Message".~~Details of the response are in [7], section 3.5.1.11, "OTAPA Response Message".~~

9.4.2.7 SECURE MODE~~Secure Mode~~

~~9.4.3.3.7~~9.4.2.7.1 Functional Description

This command corresponds to the *Secure Mode Request/Response* messages specified in [7], Sections 4.5.1.16 and 3.5.1.16, respectively.

~~9.4.3.3.8~~9.4.2.7.2 Command parameters and data

Coded as [46] Section 4.5.14, except that the CLA and INS bytes are defined in Table 1.~~The command parameters/data and response parameters/data are coded as [46] Section 4.5.14, where CLA and INS byte shall follow Section 8.1.1, and Le is the length of the data expected in response (= '01').~~

9.4.2.8 FRESH

~~9.4.3.3.9~~9.4.2.8.1 Functional Description

The function of the *FRESH* command is described in [46] Section 4.3.2.17.

~~9.4.3.3.10~~9.4.2.8.2 Command parameters and data

Coded as [46] Section 4.5.15, except that the CLA and INS bytes are defined in Table 1.~~The command parameters/data and response parameters/data are coded as [46] Section 4.5.15, where CLA and INS byte shall follow Section 8.1.1, and Le is either not present or the length of the data expected in response (= '02') depends on P1 value.~~

~~9.4.4~~9.4.3 ESN Management Commands

~~9.4.4.1~~9.4.3.1 ~~Store~~ STORE ESN_MEID_ME

~~9.4.4.1.1~~9.4.3.1.1 Functional Description

Code	Value
CLA	See Table 1~~As specified in Section 8.1.1~~
INS	See Table 1~~'DE'~~
P1	'00' if ME is not assigned with MEID_ME; '01' if ME is assigned with MEID_ME.~~See below~~
P2	'00'
Lc	'08'
Data	See below
Le	Maximum length of the response parameters data (at least 1).~~'01'~~

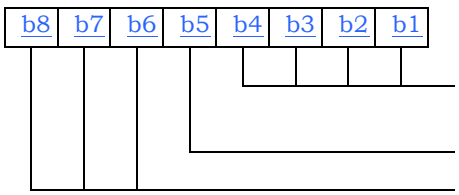
~~P1 is set to '00' if ME is assigned with ESN;~~

~~P1 is set to '01' if ME is assigned with MEID;~~

~~9.4.4.1.2~~9.4.3.1.2 Command parameters/data: (P1 = '00'):

Octet(s)	Description	Length
1	ESN_ME Length	1 byte
2 – 5~~8~~	ESN_ME	~~7~~ 4 bytes
6-8	RFU	3 bytes

Octet 1:



Length of ESN_ME ('0100').

The ME shall set this bit to '1'. The CSIM shall ignore this bit.

RFU

Octet 2 – 5:

ESN_ME is encoded with the lowest-order byte first to match the coding for $EF_{ESN\_MEID\_ME}$.~~ESN is encoded with the lowest-order byte first to match the coding for EF_{ESNME}.~~

During the ME and CSIM initialization process, the ME shall invoke the "STORE ESN_MEID_ME" command to store its ESN_ME in $EF_{ESN\_MEID\_ME}$.

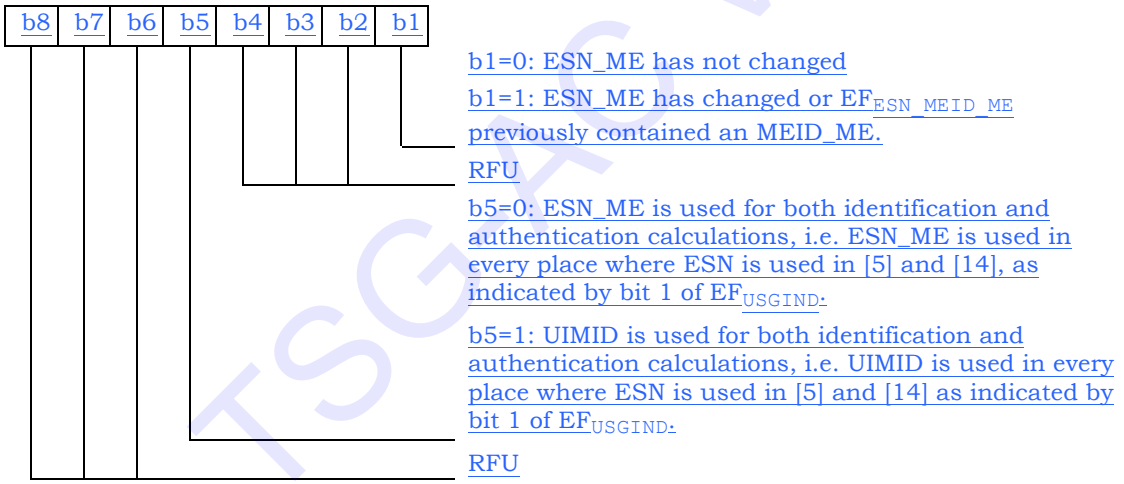
~~During the ME and CSIM initialization process, the ME shall invoke the "Store ESN_MEID_ME" command to store its ESN in EF_{ESNME} '6F38'. The ESN_ME length, expressed in octets, is specified by bits 0 through 31 through 4, inclusive of Octet 1, where bit 3 4 is MSB and bit 0 1 is LSB.~~

~~Bits 4 thru 75 through 8 of Octet 1 are RFU.~~

Response parameters/data:

Octet(s)	Description	Length
1	Change Flag, Usage Indicator	1 byte

Octet 1:



~~Bit 0 1 (LSB) of Octet 1 indicates whether the ESN_ME is different from the previous ESN or MEID that was stored in EF_{ESNME} '6F38'. Bit 0It is set to '0' if the ESN_ME has not changed and is set to '1' if it has changed.~~

~~Bits 1 through 32 through 4 are RFU are set to '000'.~~

~~Bit 4 5 of Octet 1 form a "Usage Indicator", as defined in EF 6F42EFUSGIND. Bit 4 that indicates whether the 32 LSBs of the UIM_ID or the 32 LSBs of the handset ESN are used as the "ESN" input to calculations performed using CAVE. If bit 4it is set to '1', UIM_ID is used for both identification and for authentication calculations; i.e. UIM_ID is used instead of ESN in every place where ESN is used in [5] and [14]. If bit 4it is set to '0', the handset ESN is used for both identification and for authentication calculations.~~

~~Bits 5 through 76 through 8 of Octet 1 are RFU and are set to '000'.~~

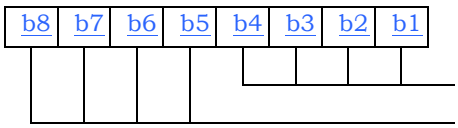
~~9.4.4.1.3~~9.4.3.1.3 Command parameters/data: (P1 = '01'): (assigned with MEID)

Octet(s)	Description	Length
1	MEID_ME Length	1 byte
2 – 8	MEID_ME	7 bytes

During the ME and CSIM initialization process, the ME shall invoke the "STORE ESN_MEID_ME" command to store its MEID_ME in $EF_{ESN\_MEID\_ME}$. ~~During the ME and CSIM initialization process, the ME shall invoke the "Store ESN_MEID_ME" command to store its MEID in EF_{ESNME} '6F38'. The MEID length, expressed in octets, is specified by bits 0 through 3~~1 through 4, inclusive, of Octet 1, ~~where bit 3~~ 4 is MSB and bit 0 1 is LSB.~~

~~Bits 4 through 7~~5 through 8 of Octet 1 ~~are RFU.~~

Octet 1:



Length of MEID_ME. This is set to '0111'.
RFU

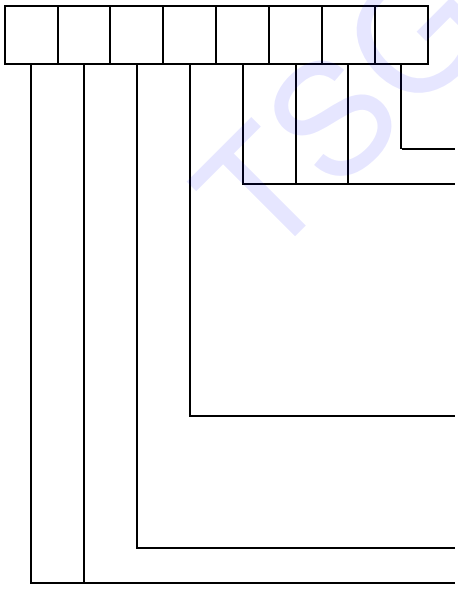
Octets 2 – 8:

MEID_ME is encoded with the lowest-order byte first to match the coding for $EF_{ESN\_MEID\_ME}$.

Response parameters/data:

Octet(s)	Description	Length
1	Change Flag, Usage Indicator	1 byte

Octet 1:



b1=0: MEID_ME has not changed
b1=1: MEID_ME has changed or $EF_{ESN\_MEID\_ME}$ previously contained an ESN_ME.
RFU
b5=0: pESN is used for both identification and authentication calculations, i.e. pESN is used in every place where ESN is used in [5] and [14], as indicated by bit 1 of EF_{USGIND}.
b5=1: UIMID is used for both identification and authentication calculations, i.e. UIMID is used in every place where ESN is used in [5] and [14] , as indicated by bit 1 of EF_{USGIND}.
b6=0: MEID_ME is used for MS identification, as indicated by bit 2 of EF_{USGIND}.
b6=1: SF_EUIMID is used for MS identification, as indicated by bit 2 of EF_{USGIND}.
RFU

~~Bit 0~~ 1 ~~(LSB)~~ of Octet 1 ~~indicates whether the MEID is different from the previous ESN or MEID that was stored in EF_{ESNME} '6F38'. Bit 0~~It is ~~set to '0' if the MEID has not changed and is set to '1' if it has changed.~~

~~Bits 1 through 3~~2 through 4 ~~are RFU and are set to '000'.~~

~~Bit 4 5 of Octet 1 forms a "Usage Indicator", as defined in EF_{USGIND} '6F42'. Bit 4 that indicates whether the 32 LSBs of the UIM_ID or the 32 LSBs of the handset Pseudo-ESN are used as the "ESN" input to calculations performed using CAVE. If bit 4it is set to '1', UIM_ID is used for both identification and for authentication calculations; i.e. UIM_ID is used instead of pseudo-ESN in every place where ESN is used in [5] and [14]. If bit 4it is set to '0', the handset Pseudo-ESN is used for both identification and for authentication calculations.~~

~~Bit 5 6 indicates whether the 56 bits of the SF_EUIMID stored in EF_{SF_EUIMID} or the 56 bits of the handset MEID is used in every place where MEID is used in [5]. If bit 5it is set to '1', then the SF_EUIMID is used. If bit 5it is set to '0', then the handset MEID is used. If service n34 is not available, b5 its value shall not be interpreted by the handset.~~

~~Bits 6 through 7 7 through 8 of Octet 1 are RFU and are set to '00'.~~

~~9.4.5~~9.4.4 Packet Data security-related Commands

9.4.4.1 COMPUTE IP AUTHENTICATION~~Compute IP Authentication~~

~~9.4.5.1.1~~9.4.4.1.1 Functional Description

This command computes responses and authenticators for use in Simple IP, Mobile IP and HRPD Access Authentication as specified in [46] Section 4.7.

~~9.4.5.1.2~~9.4.4.1.2 Command parameters and data

Coded as [46] Section 4.8.1 except that the CLA and INS bytes are defined in Table 1.~~The command parameters/data and response parameters/data are coded as [46] Section 4.8.1. where CLA and INS byte shall follow Section 8.1.1, and Le is either not present, '00', or the maximum length of the data expected in response.~~

~~9.4.6~~9.4.5 BCMCS-related Commands

9.4.5.1 BCMCS

~~9.4.6.1.1~~9.4.5.1.1 Functional Description

This command is used for BCMCS key management as specified in [46] Section 4.9 and 6.

~~9.4.6.1.2~~9.4.5.1.2 Command parameters and data

Coded as [46] Section 4.9, except that the CLA and INS bytes are defined in Table 1.~~The command parameters/data and response parameters/data are coded as [46] Section 4.9, where CLA and INS byte shall follow Section 8.1.1, and Le is either not present or the length of the data expected in response.~~

~~9.4.7~~9.4.6 Application Authentication Commands

~~9.4.7.1~~9.4.6.1 APPLICATION AUTHENTICATION~~Application Authentication~~

~~9.4.7.1.1~~9.4.6.1.1 Functional Description

The function of the APPLICATION AUTHENTICATION ~~*Application Authentication*~~ command is described in [46] Section 4.10.

~~9.4.7.1.2~~9.4.6.1.2 Command parameters and data

Coded as [46] Section 4.10, except that the CLA and INS bytes are defined in Table 1.~~The command parameters/data and response parameters/data are coded as [46] Section 4.10, where CLA and INS byte shall follow Section 8.1.1, and Le is '00' or the maximum length of the data expected in response.~~

~~9.4.8~~9.4.7 AKA-related Commands

See section 9.4.1.4 for a description of the 3G ACCESS AKA and EAP AKA AUTHENTICATE commands.~~The AKA related commands are specified in [46] Section 4.11 and 4.12, where the *3G Authentication AKA* function is specified in Section 9.4.1.4.~~

~~9.4.8.1~~9.4.7.1 UMAC GENERATION~~UMAC Generation~~

~~9.4.8.1.1~~9.4.7.1.1 Functional Description

The function of the UMAC GENERATION ~~*UMAC Generation*~~ command is described in [46] Section 4.11.

~~9.4.8.1.2~~9.4.7.1.2 Command parameters and data

Coded as [46] Section 4.12.1, except that the CLA and INS bytes are defined in Table 1.~~The command parameters/data and response parameters/data are coded as [46] Section 4.12.1, where CLA and INS byte shall follow Section 8.1.1, and Le is '00' or the maximum length of the data expected in response.~~

~~9.4.8.2~~9.4.7.2 CONFIRM_KEYS

~~9.4.8.2.1~~9.4.7.2.1 Functional Description

The function of the CONFIRM_KEYS command is described in [46] Section 4.11.

~~9.4.8.2.2~~9.4.7.2.2 Command parameters and data

Coded as [46] Section 4.12.2, except that the CLA and INS bytes are defined in Table 1.

~~The command parameters/data and response parameters/data are coded as [46] Section 4.12.2, where CLA and INS byte shall follow Section 8.1.1, and both Lc and Le are not present.~~

~~9.4.9~~9.4.8 LCS-related Commands

The command/response parameters used in this section refers to [50].

~~9.4.9.1~~9.4.8.1 S-SAFE VERIFICATION AND DECRYPTION~~Verification Decryption~~

~~9.4.9.1.1~~9.4.8.1.1 Functional Description

This command is used to verify the integrity of 'S-SAFE Envelope' and if necessary to decrypt LCS_S_SAFE_PAYLOAD afterwards. To perform integrity verification and decryption operations, the CSIM calculates a LCS_S_SAFE_KEY, a cipher key and an integrity key. For the execution of the command, the CSIM uses the LCS_ROOT_KEY, which is stored in the CSIM.

~~9.4.9.1.2~~9.4.8.1.2 Command parameters and data

Code	Value
CLA	See Table 1~~As specified in Section 8.1.1~~
INS	~~'40'~~
P1	~~'00'~~
P2	~~'00'~~
Lc	Length of command parameters/data. See below.~~See below~~
Data	See below.~~See below~~
Le	Maximum expected length of response parameters/data. See below.~~See below~~

Command parameters/data:

Octet(s)	Description	Length
1 to Lc	S-SAFE Envelope	Lc bytes

The S-SAFE Envelope formatting details are in Section 5.2.1 of [50].

Response parameters/data:

The CSIM processes the S-SAFE Envelope as described in Section 5.2.2 of [50].

If the value of LCS_S_SAFE_VERSION is not supported then CSIM shall return ~~a~~ status words SW1=~~'69'~~ '69' and SW2=~~'85'~~ '85' ("Conditions of use not satisfied")[18].

If the integrity verification has failed, then the CSIM shall return ~~a status word~~status words SW1=~~'98'~~ '98' and SW2=~~'62'~~ '62' ("Authentication error, incorrect MAC")[18].

If the integrity verification succeeds, the CSIM decrypts the LCS_S_SAFE_PAYLOAD. In such a case, the response parameters/data are:

Octet(s)	Description	Length
1 to 2	Length of LCS_S_SAFE_DATA	2 bytes
3 to ~~Le~~Nr	LCS_S_SAFE_DATA	~~Le~~Nr-2 bytes

9.4.8.2 TLS GENERATE MASTER SECRET~~Generate Master Secret~~

~~9.4.9.1.3~~9.4.8.2.1 Functional Description

This command is used to generate the *master_secret* as described in Section 5.3.8.1 of [50]. The CSIM will assign a *master_secret_index* for each generated *master_secret*. CSIM shall securely store the *master_secret* and its corresponding *master_secret_index*, and shall only return the *master_secret_index* to the ME.

In order to generate the *master_secret*, CSIM first calculates the LCS_UIM_HPS_TLS_PSK_KEY for TLS Session-A; or LCS_UIM_PDE_ROOT_KEY and LCS_UIM_PDE_TLS_PSK_KEY for TLS Session-B. For the execution of the command, the CSIM uses the LCS_ROOT_KEY, which is stored in the CSIM.

~~9.4.9.1.4~~9.4.8.2.2 Command parameters and data

Code	Value
CLA	See Table 1~~As specified in Section 8.1.1~~
INS	~~'42'~~
P1	~~'00'~~
P2	(See Detail 1)
Lc	Length of the command parameters/data. See below~~See below~~
Data	See below
Le	Maximum length of the response parameters/data. See below~~See below~~

Detail 1:

If DHE Key exchange is used, then the resulting *other_secrets* parameter (equal to the shared secret DH key) inside the data field parameter is so large that it is possible to have Lc exceeds 254 bytes. Therefore, this command shall chain successive blocks of with a maximum size of 254 bytes each. If the blocks used within the command are received ~~run~~ out of sequence, the card shall return, SW1='98' and SW2='34'.[17]

P2 contains chaining information as follows:

b8	b7	b6	b5	b4	b3	b2	b1	Meaning
0	0	0	0	0	0	0	0	First block
X	X	X	X	0	0	0	1	'xxxx' indicates (n+1)th next block. '0000 0001' = 1st next block. '0001 0001' = 2nd next block. '0010 0001' = 3rd next block. … '1111 0001' = 16th next block.
0	0	0	0	0	0	1	0	Single block
0	0	0	0	0	0	1	1	Last block

* Le: 'Not present' for P2 = '00' or 'x1'

At least 16 ~~bytes~~ for P2 = ‘02’ or ‘03’

Command parameters/data:

a. Operation for TLS Session-A (SessionType=’01’)

Octet(s)	Description	Length
1	TLS Service Type (see Table 2)	1 byte
2	SessionType	1 byte
3 to A+2	TLS Server_Version TLV	A bytes
A+3 to A+B+2	TLS Other_Secret TLV	B bytes
A+B+3 to A+B+C+2	TLS Master_Client_Random TLV	C bytes
A+B+C+3 to A+B+C+D+2	TLS Master_Server_Random TLV	D bytes
NOTE: The tags inside TLV objects in the command are specified in Annex D of this document.		

The coding for 'TLS Service Type' is defined according to the following table:

~~Table 7~~ Table 2 Coding of 'TLS Service Type'

Binary Value	Service Type
‘00000000’	IP-based Location Services
Others	Reserved

For "IP-based Location Services" (i.e. 'TLS Service Type' = ‘0x00’), see [50] for the definition of the remaining input parameters.

b. Operation for TLS Session-B (SessionType=’02’)

Octet(s)	Description	Length
1	TLS Service Type (Table 2)	1 byte
2	SessionType	1 byte
2 to A+2	TLS PSK VERSION TLV	A bytes
A+3 to A+B+2	TLS PSK EXPIRY TLV	B bytes
A+B+3 to A+B+C+2	TLS PSK RAND TLV	C bytes
A+B+C+3 to A+B+C+D+2	TLS PSK EXTRAS TLV	D bytes
A+B+C+D+3 to A+B+C+D+2	TLS Server_Version TLV	E bytes
A+B+C+D+E+3 to A+B+C+D+E+F+2	TLS Other_Secret TLV	F bytes
A+B+C+D+E+F+3 to A+B+C+D+E+F+G+2	TLS Master_Client_Random TLV	G bytes
A+B+C+D+E+F+G+3 to A+B+C+D+E+F+G+H+2	TLS Master_Server_Random TLV	H bytes
NOTE: The tags inside TLV objects in the command are specified in Annex D of this document.		

Response parameters/data:

Octet(s)	Description	Length
1 to 2	*master_secret_index*	2 bytes

9.4.8.3 TLS GENERATE VERIFY DATA~~Generate Verify Data~~

~~9.4.9.1.5~~9.4.8.3.1 Functional Description

This command is used to generate both TLS Session-A and TLS Session-B client's *verify_data*, as described in [50].

~~9.4.9.1.6~~9.4.8.3.2 Command parameters and data

Code	Value
CLA	~~As specified in Section 8.1.1~~
INS	~~'44'~~
P1	~~'00'~~
P2	~~'00'~~See Table 1
Lc	Length of the command parameters/data. See below
Data	See below
Le	Maximum length of the response parameters/data. See below

Command parameters/data:

Octet(s)	Description	Length
1	TLS Service Type (see Table 2)	1 byte
2 to 3	TLS Master_Secret_Index TLV	2 bytes
4 to A+3	TLS MS Verify_Digest TLV	A bytes
NOTE: The tags inside TLV objects in the command are specified in Annex D of this document.		

Response parameters/data:

Octet(s)	Description	Length
1-2	MS Verify Data Length	2 bytes
3 to B+2	MS Verify Data	B bytes

9.4.8.4 TLS VERIFICATION AND GENERATE KEY BLOCK ~~TLS Verify Data & Generate Key Block~~

~~9.4.9.1.7~~9.4.8.4.1 Functional Description

This command is used to verify the Server's *verify_data* from the server (HPS or PDE) during TLS Session-A or TLS Session-B handshake, and then generates the *key_block* data, as described in [50].

~~9.4.9.1.8~~9.4.8.4.2 Command parameters and data

Code	Value
CLA	~~As specified in Section 8.1.1~~
INS	~~'46'~~
P1	~~'00'~~
P2	~~'00'~~See Table 1
Lc	Length of the command parameters/data. See below
Data	See below
Le	Maximum length of the response parameters/data. See below

Command parameters/data:

Octet(s)	Description	Length
1	TLS Service Type (see Table 2)	1 byte
2 to A+1	TLS Server_Version TLV	A bytes
A+2 to A+B+1	TLS Master_Secret_Index TLV	B bytes
A+B+2 to A+B+C+1	TLS Current_Client_Random TLV	C bytes
A+B+C+2 to A+B+C+D+1	TLS Current_Server_Random TLV	D bytes
A+B+C+D+2 to A+B+C+D+E+1	TLS Server Verify_Digest TLV	E bytes
A+B+C+D+E+2 to A+B+C+D+E+F+1	TLS Server Verify_Data TLV	F bytes
A+B+C+D+E+F+2 to A+B+C+D+E+F+3	TLS Key_Block_Len	2 bytes
NOTE: The tags inside TLV objects in the command are specified in Annex D of this document.		

Response parameters/data:

Octet(s)	Description	Length
1-2	TLS *key_block* Length	2 bytes
3 to G+2	TLS *key_block*	G bytes

If the verification fails, the CSIM shall return a status word SW1=~~'98'~~ '98' and SW2=~~'62'~~'62' ~~("~~("Authentication error~~")~~") [see section 4.2].

10. DESCRIPTION OF SERVICES-RELATED PROCEDURE

10.1 IP-based Location Services Procedures [50]

10.1.1 Functionalities of CSIM and ME

10.1.1.1 CSIM

- Generate LCS_UIM_S_SAFE Key, LCS_UIM_HPS_TLS_PSK Key and LCS_UIM_PDE_ROOT Key from LCS Root Key. This may be done at the same time when LCS Root Key is provisioned or may be later.
- Generate LCS_S_SAFE_CK and LCS_S_SAFE_IK from LCS_UIM_S_SAFE Key after receiving the ~~'S~~'S-SAFE VERIFICATION AND DECRYPTION' ~~Verification and Decryption'~~ command from the ME, and
- perform Integrity Verification to LCS_S_SAFE_MAC_DATA with LCS_S_SAFE_IK, and
- when necessary, decrypt LCS_S_SAFE_PAYLOAD with LCS_S_SAFE_CK.
- Compute *master_secret* with input parameters after receiving the 'TLS GENERATE MASTER SECRET~~Generate Master Secret~~' command from the ME, assign a unique 16-bit *master_secret_index* for the calculated *master_secret*.
- Compute Session-A (or Session-B) *verify_data* with input parameters after receiving the 'TLS GENERATE VERIFY DATA ~~Generate verify_data~~' command from the ME.
- Verify the received H-PS (or PDE) Verify Data and if success then generate a *key_block* from inputs parameters after receiving the 'TLS VERIFICATION AND GENERATE KEY BLOCK ~~Verify data and Generate key_block~~' command from the ME.

10.1.1.2 ME

- Perform Expiry Check and Replay Detection against S-SAFE envelop
- Generate MS Verify Digest.
- Generate MS *session_secret*.
- Perform bulk ciphering and integrity check for TLS Session-A application data with Session-A Session Secret
- Perform bulk ciphering and integrity check for TLS Session-B application data with Session-B Session Secret
- Issue correct command with appropriate parameters to CSIM.

10.1.2 Key Management

If service n17 is available, these following keys shall be securely maintained in the CSIM:

- LCS_ROOT_KEY.

- three PSK keys (i.e. LCS_UIM_S_SAFE Key, LCS_UIM_HPS_TLS_PSK Key and LCS_UIM_PDE_ROOT Key) derived from LCS_ROOT_KEY.
- *master_secret* and *master_secret_index*

When ME sends a ~~'TLS Generate Master Secret~~TLS GENERATE MASTER SECRET' command for TLS Session-B, the CSIM shall generate a LCS_UIM_PDE_TLS_PSK_KEY from LCS_UIM_PDE_ROOT_KEY and the input parameter LCS_UIM_PDE_TLS_PSK_RAND. LCS_UIM_PDE_TLS_PSK_KEY (not the LCS_UIM_PDE_ROOT Key) shall then be used to generate the requested *master_secret*.

ANNEX A (INFORMATIVE): R-UIM/CSIM FILE MAPPING TABLE

The following section provides some guidelines for file mapping between an R-UIM and CSIM in a UICC. It should be noted that some files are optional, and these files are not necessarily present in the R-UIM or CSIM application. Mapping with multiple CSIM's is not considered.

1. Files mapped between an R-UIM and a CSIM should be of the same size.

2. If subscription related information is different across an R-UIM and a CSIM, the files cannot be mapped.

3. Mapping is not possible if the file is applicable only either to an R-UIM or a CSIM, e.g. $EF_{Revision}$.

4. Case by case analysis has to be done by the network operators/card manufacturers for files to be mapped that are specific to the terminal, e.g. ESN, MEID files, etc that contains device specific information.

ANNEX B (NORMATIVE): LIST OF SFI VALUES

~~List of SFI Values~~

File Identification	SFI	Description
'6F43'	'01'	Administrative data
'6F32'	'02'	CSIM Service Table
'6F2C'	'03'	Access Overload Class
'6F22'	'04'	IMSI_M
'6F23'	'05'	IMSI_T
'6F24'	'06'	TMSI
'6F30'	'07'	PRL
'6F41'	'08'	CDMA Home Service Provider ~~Display Information~~Name
'6F47'	'09'	Emergency Call Codes
'6F3A'	'0A'	Language Indication
'6F6B'	'0B'	3G Cipher and Integrity Key
'6F28'	'0C'	CDMA Home SID and NID
'6F2A'	'0D'	CDMA System-Network Registration Indicators
'6F5A'	'0E'	Extended PRL
'6F75'	'0F'	Enabled Services Table
'6F7C'	'10'	Incoming Call Information
'6F7D'	'11'	Outgoing Call Information
'6F7F'	'12'	Capability Control Parameters2

ANNEX C (INFORMATIVE): CSIM APPLICATION SESSION ACTIVATION/TERMINATION

The purpose of this annex is to illustrate the different Application Session procedures.

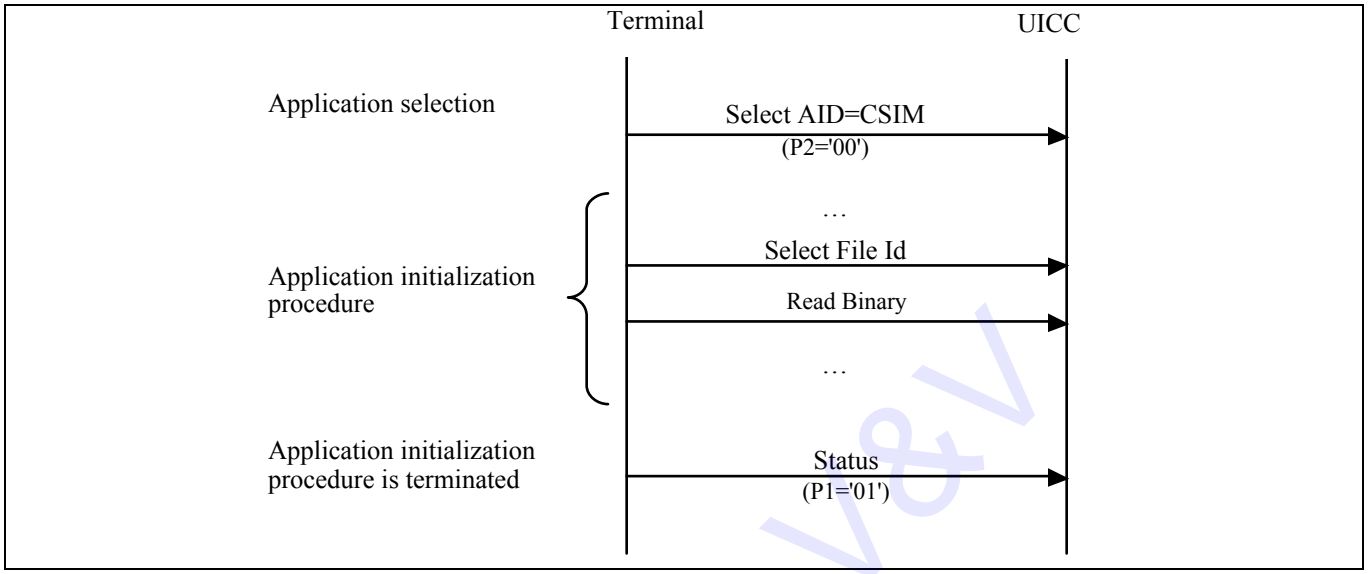


Figure 1 CSIM Application Session Activation Procedures

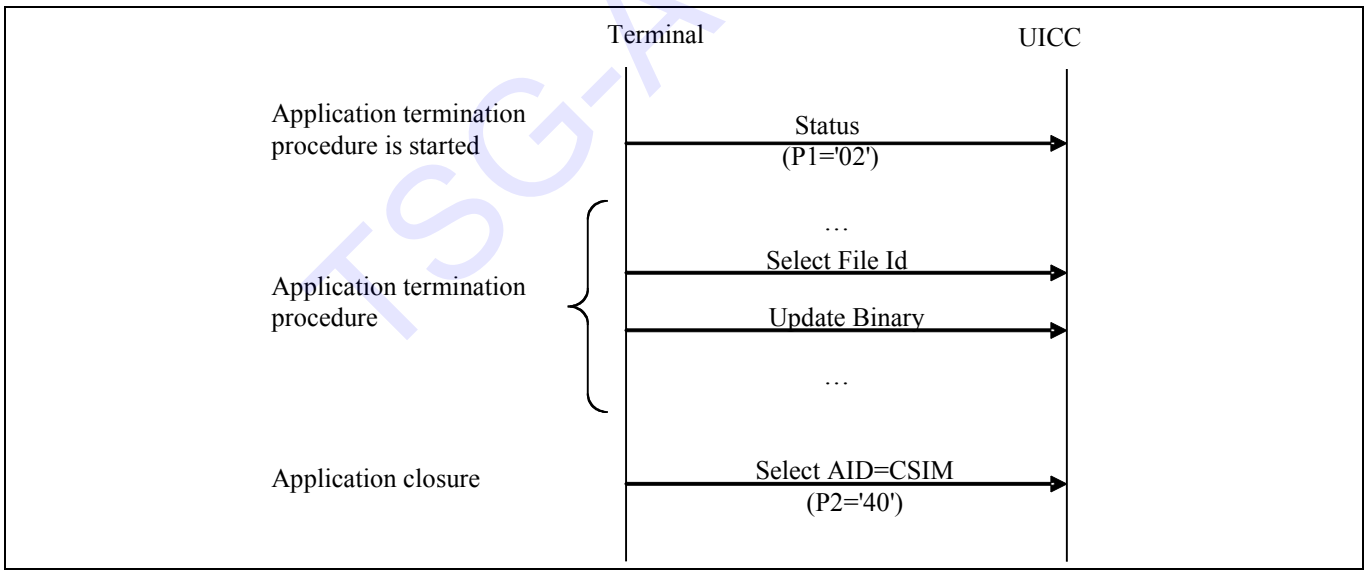


Figure 2 CSIM Application Session Termination Procedures

ANNEX D (NORMATIVE): TLS-RELATED TAG VALUES

Tag	Name of Data Element	Usage
'80'	TLS Server_Version TLV objects	TLS command
'81'	TLS Cipher_Suite TLV objects	TLS command
'82'	TLS Other_Secret TLV object	TLS command
'83'	TLS Master_Client_Random TLV object	TLS command
'84'	TLS Master_Server_Random TLV object	TLS command
'85'	TLS Current_Client_Random TLV object	TLS command
'86'	TLS Current_Server_Random TLV object	TLS command
'87'	TLS Server Verify_Digest TLV object	TLS command
'88'	TLS Server Verify_Data TLV object	TLS command
'89'	TLS MS Verify_Digest TLV object	TLS command
'8A'	TLS_Master_Secret_Index TLV object	TLS command
'8B'	TLS PSK VERSION TLV	TLS command
'8C'	TLS PSK EXPIRY TLV	TLS command
'8D'	TLS PSK RAND TLV	TLS command
'8E'	TLS PSK EXTRAS TLV	TLS command

ANNEX E (INFORMATIVE): SUGGESTED CONTENTS OF THE EFS AT PRE-PERSONALIZATION

Table 3 is a general outline of the CSIM files defined in this specification.

1. All values are sized in bytes unless otherwise noted.
2. Default Values are specified when available and are intended to be guidelines only. In some cases, operators must specify explicit parameter values as no logical default exists. In the case where the parameter values are necessary, valid values and/or ranges are listed.
3. Default and Parameter values are for general quick reference only and not intended to specify details. Refer to the corresponding file for details.
4. Default Values and Parameter Values are specified in Hexadecimal, unless otherwise noted.
5. GSM-specific files are not included.
6. If EFs have an unassigned value, it may not be clear from the main text what this value should be. This annex suggests values in these cases.
7. File sizes are in bytes.
8. Mandatory files are indicated as 'M' and optional files by 'O'.

~~Table 8~~ Table 3 Summary of CSIM Files

File Name	File ID – File Type	Access Read – Update	Access – Invalidate-Rehabilitate	Size – M/O	Default Values (D) and/or Parameter Values (P) in Bytes
Authentication – NAM Parameters and Operational Parameters					
A-Key	––	Never–Never	-	8–M	Specified by Operator
Root Key	––	Never–Never	-	16–M	Specified by Operator
BCMCS Root Key	––	Never–Never	-	16–O	Specified by Operator
IMS Root Key	––	Never–Never	-	16–O	Specified by Operator
WLAN Root Key	––	Never–Never	-	16–O	Specified by Operator
SSD	––	Never–Never	-	16–M	-
EFCOUNT	6F21–CY	PIN–PIN	ADM-ADM	2–M	D = '00 00'
BAK	––	Never–Never	-	16–O	Specified by Operator

File Name	File ID – File Type	Access Read – Update	Access – Invalidate- Rehabilitate	Size – M/O	Default Values (D) and/or Parameter Values (P) in Bytes
UpdatedBAK	----	Never–Never	-	16–O	Specified by Operator
SharedSecret	----	Never–Never	-	Variable–O	Specified by Operator
UAK	----	Never–Never	-	16–O	Specified by Operator
SQNMS	----	Never–Never	-	6–O	-
NAM Parameters and Operational Parameters					
EF_{IMSI_M}	6F22–TR	PIN–ADM	ADM-PIN	10–M	P = Specified by Operator or D='00…00'
EF_{IMSI_T}	6F23–TR	PIN–ADM	ADM-PIN	10–M	P = Specified by Operator or D='00…00'
EF_{TMSI}	6F24–TR	PIN–PIN	ADM-PIN	16–M	D = '00 00 00 00 00 00 00 00 FF FF FF FF 00 00 00'
EF_{AH}	6F25–TR	PIN–PIN	ADM-ADM	2–M	P = Specified by Operator or D = '00 00'
EF_{AOP}	6F26–TR	PIN–PIN	ADM-ADM	1–M	-
EF_{ALOC}	6F27–TR	PIN–PIN	ADM-ADM	7–M	-
$EF_{CDMAHOME}$	6F28–LF	PIN–PIN	ADM-ADM	5–M	P = Specified by Operator or D = '00 00 00 00 00'
EF_{ZNREGI}	6F29–LF	PIN–PIN	ADM-ADM	8–M	D = '00 00 00 00 00 00 00 00'
EF_{SNREGI}	6F2A–TR	PIN–PIN	ADM-ADM	7–M	-
$EF_{DISTREGI}$	6F2B–TR	PIN–PIN	ADM-ADM	8–M	D = '00 00 00 00 00 00 00 00'
EF_{ACCOLC}	6F2C–TR	PIN–ADM	ADM-ADM	1–M	P = '00' to '0F' derived from IMSI_M / IMSI_T
EF_{TERM}	6F2D–TR	PIN–PIN	ADM-ADM	1–M	Specified by Operator P = '00' to '07'
EF_{SSCI}	6F2E–TR	PIN–PIN	ADM-ADM	1–O	Specified by Operator P = '00' to '07'
EF_{ACP}	6F2F–TR	PIN–PIN	ADM-ADM	7–M	Specified by Operator
EF_{PRL}	6F30–TR	PIN–ADM	ADM-ADM	Variable–M	Specified by Operator
EF_{RUIMID}	6F31–TR	ALW–~~NEVER~~Never	~~NEVER~~Never-~~NEVER~~Never	8–M	Specified by CSIM Manufacturer
EF_{CSIM_ST}	6F32–TR	PIN–ADM	ADM-ADM	Variable–M	Specified by Operator

File Name	File ID – File Type	Access Read – Update	Access – Invalidate-Rehabilitate	Size – M/O	Default Values (D) and/or Parameter Values (P) in Bytes
EF_{SPC}	6F33–TR	ADM–ADM	ADM-ADM	3–M	D = '00 00 00' or P = '00 00 00' to '99 99 99'
$EF_{OTAPASPC}$	6F34–TR	PIN–PIN	ADM-ADM	1–M	Specified by Operator or D = '00'
$EF_{NAMLOCK}$	6F35–TR	PIN–PIN	ADM-ADM	1–M	Specified by Operator
EF_{OTA}	6F36–TR	PIN–ADM	ADM-ADM	Variable–M	P = Defined in [7]
EF_{SP}	6F37–TR	PIN–PIN	ADM-ADM	1–M	Specified by Operator
$EF_{ESNMEESN\_MEID\_ME}$	6F38–TR	ALW–ADM	ADM-ADM	8–M	D ='00…00'
EF_{LI}	6F3A–TR	ALW–PIN	ADM-ADM	Variable–M	D = 'FF… FF'
EF_{FDN}	6F3B–LF	PIN–PIN2	ADM-ADM	Variable-O	D = 'FF…FF'
EF_{SMS}	6F3C–LF	PIN–PIN	ADM-ADM	Variable–O	D = '00 FF…FF'
EF_{SMSP}	6F3D–LF	PIN–PIN	ADM-ADM	Variable–O	D = 'FF…FF'
EF_{SMSS}	6F3E–TR	PIN–PIN	ADM-ADM	Variable–O	D = 'FF…FF'
EF_{SSFC}	6F3F–TR	PIN–PIN	ADM-ADM	Variable–O	Specified by Operator
EF_{SPN}	6F41–TR	ALW–ADM	ADM-ADM	35–O	Specified by Operator
EF_{USGIND}	6F42–TR	PIN–ADM	ADM-ADM	1–M	Specified by Operator
EF_{AD}	6F43–TR	ALW–ADM	ADM-ADM	Variable–M	D = '00…00'
EF_{MDN}	6F44–LF	PIN–PIN	ADM-ADM	11–O	Specified by Operator
EF_{MAXPRL}	6F45–TR	PIN–ADM	ADM-ADM	2 or 4–M	Specified by Operator
EF_{SPCS}	6F46–TR	PIN–NEVERNever	NEVERNever-NEVERNever	1–M	P = If EF 6F33 is set to default value then D = '00' otherwise D = '01'
EF_{ECC}	6F47–TR	ALW–ADM	ADM-ADM	Variable–O	D = 'FF'
$EF_{ME3GPDOPC}$	6F48–TR	PIN–PIN	ADM-ADM	1–O	D = '00'
$EF_{3GPDOPM}$	6F49–TR	PIN–ADM	ADM-ADM	1–O	Specified by Operator
EF_{SIPCAP}	6F4A–TR	PIN–ADM	ADM-ADM	4–O	Specified by Operator
EF_{MIPCAP}	6F4B–TR	PIN–ADM	ADM-ADM	5–O	Specified by Operator
EF_{SIPUPP}	6F4C–TR	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
EF_{MIPUPP}	6F4D–TR	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
EF_{SIPSP}	6F4E–TR	PIN–PIN	ADM-ADM	1–O	Specified by Operator
EF_{MIPSP}	6F4F–TR	PIN–PIN	ADM-ADM	Variable–O	Specified by Operator

File Name	File ID – File Type	Access Read – Update	Access – Invalidate-Rehabilitate	Size – M/O	Default Values (D) and/or Parameter Values (P) in Bytes
$EF_{SIPPAPSS}$	6F50–TR	PIN–PIN	ADM-ADM	Variable–O	Specified by Operator
SimpleIP CHAP SS	----	Never–Never	-	Variable–O	Specified by Operator
MobileIP SS	----	Never–Never	-	Variable–O	Specified by Operator
Shared Secret	----	Never–Never	-	Variable–O	Specified by Operator
EF_{PUZL}	6F53–TR	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
$EF_{MAXPUZL}$	6F54–TR	PIN–ADM	ADM-ADM	5–O	Specified by Operator
EF_{MECRP}	6F55–TR	PIN–PIN	ADM-ADM	3–M	D = '00 00 00'
$EF_{HRPDCAP}$	6F56–TR	PIN–ADM	ADM-ADM	2–O	Specified by Operator
$EF_{HRPDUPP}$	6F57–TR	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
HRPD AA CHAP SS	----	Never–Never	-	Variable–O	Specified by Operator
EF_{CSSPR}	6F58–TR	PIN–ADM	ADM-ADM	1–O	D = 'FF'
EF_{ATC}	6F59–TR	PIN–ADM	ADM-ADM	1–O	Specified by Operator
EF_{EPRL}	6F5A–TR	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
$EF_{BCSMScfg}$	6F5B–TR	PIN–ADM	ADM-ADM	1–O	Specified by Operator
$EF_{BCSMSpref}$	6F5C–TR	PIN–PIN	ADM-ADM	1–O	D = 'FF'
$EF_{BCSMStable}$	6F5D–LF	PIN–ADM	ADM-ADM	Variable–O	D = '00 FF…FF'
EF_{BCSMSp}	6F5E–LF	PIN–PIN	ADM-ADM	2–O	D = 'FF FF'
$EF_{BAKPARA}$	6F63–LF	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
$EF_{UpBAKPARA}$	6F64–CY	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
EF_{MMSN}	6F65–LF	PIN–PIN	ADM-ADM	Variable–O	D='00 00 00 FF…FF'
EF_{EXT8}	6F66–LF	PIN–PIN	ADM-ADM	Variable–O	D='FF…FF'
EF_{MMSICP}	6F67–TR	PIN–ADM	ADM-ADM	Variable–O	D='FF...FF'
EF_{MMSUP}	6F68–LF	PIN–PIN	ADM-ADM	Variable–O	D='FF...FF'
EF_{MMSUCP}	6F69–TR	PIN–PIN/PIN2	ADM-ADM	Variable–O	D= 'FF…FF'
$EF_{AuthCapability}$	6F6A–LF	PIN–ADM	ADM-ADM	Variable–O	D= '00…00'
EF_{3GCIK}	6F6B–TR	PIN–ADM	ADM-ADM	32–O	Specified by Operator
EF_{DCK}	6F6C–TR	PIN–PIN	ADM-ADM	20–O	Specified by Operator
EF_{GID1}	6F6D–TR	PIN–ADM	ADM-ADM	N–O	Specified by Operator

File Name	File ID – File Type	Access Read – Update	Access – Invalidate- Rehabilitate	Size – M/O	Default Values (D) and/or Parameter Values (P) in Bytes
EF_{GID2}	6F6E–TR	PIN–ADM	ADM-ADM	N–O	Specified by Operator
$EF_{CDMACNL}$	6F6F–TR	PIN–ADM	ADM-ADM	7N–O	Specified by Operator
EF_{HOME_TAG}	6F70–TR	PIN–ADM	ADM-ADM	N–~~O~~M	Specified by Operator
EF_{GROUP_TAG}	6F71–TR	PIN–ADM	ADM-ADM	Variable–~~O~~M	Specified by Operator
$EF_{SPECIFIC\_TAG}$	6F72–TR	PIN–ADM	ADM-ADM	Variable–~~O~~M	Specified by Operator
EF_{CALL_PROMPT}	6F73–TR	PIN–ADM	ADM-ADM	Variable–~~O~~M	Specified by Operator
EF_{SF_EUIMID}	6F74–TR	ALW–~~NEVER~~Never	~~NEVER~~Never-~~NEVER~~Never	7–O	Specified by CSIM Manufacturer
EF_{EST}	6F75–TR	PIN–PIN	ADM-ADM	Variable–O	Specified by Operator
$EF_{~~HIDDEN\_KEY~~HiddenKey}$	6F76–TR	PIN–ADM	ADM-ADM	–O	Specified by Operator
EF_{LCSVER}	6F77–TR	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
EF_{LCSCP}	6F78–TR	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
EF_{SDN}	6F79–LF	PIN–PIN2	ADM-ADM	Variable–O	Specified by Operator
EF_{EXT2}	6F7A–LF	PIN–ADM	ADM-ADM	13–O	Specified by Operator
EF_{EXT3}	6F7B–LF	PIN–PIN	ADM-ADM	13–O	Specified by Operator
EF_{ICI}	6F7C–CY	PIN–PIN	ADM-ADM	Variable–O	Specified by Operator
EF_{OCI}	6F7D–CY	PIN–PIN	ADM-ADM	Variable–O	Specified by Operator
EF_{EXT5}	6F7E–LF	PIN–PIN	ADM-ADM	13–O	Specified by Operator
EF_{CCP2}	6F7F–LF	PIN–PIN	ADM-ADM	Variable–O	~~Specified by Operator~~D='FF…FF'
$EF_{AppLabels}$	6F80–TR	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
EF_{Model}	6F81–TR	PIN–PIN	ADM-ADM	126–O	D='FF…FF'
EF_{RC}	6F82–TR	ALW–ADM	ADM-ADM	Variable–O	Specified by Operator
EF_{SMSCAP}	6F83–TR	PIN–ADM	ADM-ADM	4–O	Specified by Operator
$EF_{MIPFlags}$	6F84–TR	PIN–ADM	ADM-ADM	1–O	Specified by Operator
~~$EF_{SIPUPPExt}$~~$EF_{3GPDUPPExt}$	6F85–TR	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
~~$EF_{MIPUPPExt}$~~Reserved	6F86–TR	~~PIN–ADM~~n/a	~~ADM-ADM~~n/a	~~Variable–O~~n/a	~~Specified by Operator~~n/a

File Name	File ID – File Type	Access Read – Update	Access – Invalidate-Rehabilitate	Size – M/O	Default Values (D) and/or Parameter Values (P) in Bytes
$EF_{IPV6CAP}$	6F87–TR	PIN–ADM	ADM-ADM	21–O	Specified by Operator
$EF_{TCPConfig}$	6F88–TR	PIN–ADM	ADM-ADM	2–O	Specified by Operator
EF_{DGC}	6F89–TR	PIN–ADM	ADM-ADM	3–O	Specified by Operator
$EF_{WAPBrowserCP}$	6F8A–TR	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator
$EF_{WAPBrowserBM}$	6F8B–TR	PIN–PIN	ADM-ADM	Variable–O	D='FF…FF'
$EF_{MMSConfig}$	6F8C–TR	PIN–ADM	ADM-ADM	8–O	Specified by Operator
EF_{JDL}	6F8D–TR	PIN–ADM	ADM-ADM	Variable–O	Specified by Operator

1

ANNEX F (NORMATIVE): RESERVATION OF FILE IDENTIFIERS

The following FIDs are reserved by the present document:

- ADF:
  - Operational use (implicit FID for the current ADF):
    - '7FFF'.
- Dedicated Files:
  - Administrative use:
    - '7F4X', '5F1X', '5F2X'.
  - Operational use:
    - '7F10' ($DF_{TELECOM}$), '7F20' (DF_{GSM}), '7F21' ($DF_{DCS1800}$), '7F22' (DF_{IS-41}), '7F23' (DF_{FP-CTS}).
      - Reserved under '7F10' ($DF_{TELECOM}$):
        - '5F50' ($DF_{GRAPHICS}$); '5F3A' ($DF_{PHONEBOOK}$); '5F3B' ($DF_{MULTIMEDIA}$); '5F3C' (DF_{MMSS}).
    - '7F24' ($DF_{TIA/EIA-136}$), '7F25' ($DF_{TIA/EIA-95}$) and '7F2X', where X ranges from '6' to 'F'.
    - '7F80' (DF_{PDC}) is used for the Japanese PDC specification.
    - '7F90' (DF_{TETRA}) is used for the TETRA specification.
    - '7F31' (DF_{IDEN}) is used in the iDEN specification.
- Elementary files:
  - Administrative use:
    - '6F XX' in the DFs '7F 4X'; '4F XX' in the DFs '5F 1X', '5F2X'.
    - '6F 1X' in the DFs '7F 10', '7F 20', '7F 21';
    - '4F 1X' in all 2nd level DFs;
    - '2F EX' in the MF '3F 00'.
  - Operational use:
    - '6F 2X', '6F 3X', '6F 4X' in '7F 10' and '7F 2X';
    - '4F YX', where Y ranges from '2' to 'F' in all 2nd level DFs;
    - '2F05', '2F06' and '2F 1X' in the MF '3F 00'.
  - Operational use ~~ISO/IEC 7816-4 [12]~~[53]:
    - '2F00' EF_{DIR}, '2F01' EF_{ATR} in the MF '3F00'.
  - Reserved under CDMA ADF:
    - '6F86': Reserved
    - From '6F8E' to '6F96' (reserved for CDG)

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