

# ETSI TS 102 223 V11.3.0 (2014-06)



Technical Specification

**Smart Cards;  
Card Application Toolkit (CAT)  
(Release 11)**

---

Reference

RTS/SCP-T003r08avb30

---

Keywords

smart card

**ETSI**

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

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

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

---

**Important notice**

The present document can be downloaded from:

<http://www.etsi.org>

The present document may be made available in electronic versions and/or in print. The content of any electronic and/or print versions of the present document shall not be modified without the prior written authorization of ETSI. In case of any existing or perceived difference in contents between such versions and/or in print, the only prevailing document is the print of the Portable Document Format (PDF) version kept on a specific network drive within ETSI Secretariat.

Users of the present document should be aware that the document may be subject to revision or change of status. Information on the current status of this and other ETSI documents is available at

<http://portal.etsi.org/tb/status/status.asp>

If you find errors in the present document, please send your comment to one of the following services:

[http://portal.etsi.org/chaicor/ETSI\\_support.asp](http://portal.etsi.org/chaicor/ETSI_support.asp)

---

**Copyright Notification**

No part may be reproduced or utilized in any form or by any means, electronic or mechanical, including photocopying and microfilm except as authorized by written permission of ETSI.

The content of the PDF version shall not be modified without the written authorization of ETSI.

The copyright and the foregoing restriction extend to reproduction in all media.

© European Telecommunications Standards Institute 2014.

All rights reserved.

**DECT™**, **PLUGTESTS™**, **UMTS™** and the ETSI logo are Trade Marks of ETSI registered for the benefit of its Members. **3GPP™** and **LTE™** are Trade Marks of ETSI registered for the benefit of its Members and of the 3GPP Organizational Partners.

**GSM®** and the GSM logo are Trade Marks registered and owned by the GSM Association.

# Contents

Intellectual Property Rights .....	11
Foreword.....	11
Modal verbs terminology.....	11
1 Scope .....	12
2 References .....	12
2.1 Normative references .....	13
2.2 Informative references.....	15
3 Definitions, symbols and abbreviations .....	15
3.1 Definitions.....	15
3.2 Symbols.....	17
3.3 Abbreviations .....	17
4 Overview of CAT.....	19
4.1 Profile download .....	19
4.2 Proactive UICC .....	19
4.3 Data download to UICC.....	20
4.4 Menu selection .....	20
4.5 Call control by network access application .....	20
4.6 Void.....	20
4.7 Event download.....	20
4.8 Security .....	20
4.9 Multiple card .....	20
4.10 Timer expiration.....	21
4.11 Bearer Independent Protocol.....	21
4.12 Description of the access technology indicator mechanism .....	21
4.13 Tag allocation guidelines.....	21
4.14 Description of the network search mode mechanism .....	22
4.15 CAT operation in reduced capability terminals.....	22
4.16 CAT over the modem interface .....	22
4.17 CAT facilities provided by eCAT clients .....	23
5 Profile download .....	24
5.1 Procedure.....	24
5.2 Structure and coding of TERMINAL PROFILE.....	24
5.3 Definition of display parameters in profile download .....	31
5.3.1 Number of characters supported down the terminal display.....	31
5.3.2 Number of characters supported across the terminal display.....	31
5.3.3 Display can be resized .....	32
5.3.4 Text wrapping.....	32
5.3.5 Text scrolling.....	32
5.3.6 Width reduction when in a menu .....	32
5.3.7 Text attributes .....	32
6 Proactive UICC .....	33
6.1 Introduction .....	33
6.2 Identification of terminal support.....	35
6.3 General procedure .....	36
6.4 Proactive UICC commands and procedures.....	36
6.4.1 DISPLAY TEXT .....	36
6.4.2 GET INKEY .....	38
6.4.3 GET INPUT.....	39
6.4.4 MORE TIME.....	40
6.4.5 PLAY TONE .....	41
6.4.6 POLL INTERVAL .....	42
6.4.7 REFRESH.....	42

6.4.8	SET UP MENU .....	44
6.4.9	SELECT ITEM .....	44
6.4.10	SEND SHORT MESSAGE .....	45
6.4.11	Void .....	46
6.4.12	Void .....	46
6.4.13	SET UP CALL .....	46
6.4.14	POLLING OFF .....	48
6.4.15	PROVIDE LOCAL INFORMATION .....	48
6.4.16	SET UP EVENT LIST .....	48
6.4.17	PERFORM CARD APDU .....	49
6.4.18	POWER OFF CARD .....	49
6.4.19	POWER ON CARD .....	50
6.4.20	GET READER STATUS .....	50
6.4.21	TIMER MANAGEMENT .....	51
6.4.22	SET UP IDLE MODE TEXT .....	51
6.4.23	RUN AT COMMAND .....	52
6.4.24	SEND DTMF .....	52
6.4.25	LANGUAGE NOTIFICATION .....	53
6.4.26	LAUNCH BROWSER .....	53
6.4.27	OPEN CHANNEL .....	54
6.4.27.1	OPEN CHANNEL related to CS bearer .....	54
6.4.27.2	OPEN CHANNEL related to packet data service bearer .....	56
6.4.27.3	OPEN CHANNEL related to local bearer .....	57
6.4.27.4	OPEN CHANNEL related to Default (network) Bearer .....	59
6.4.27.5	OPEN CHANNEL related to UICC Server Mode .....	60
6.4.27.6	OPEN CHANNEL related to Terminal Server Mode .....	61
6.4.28	CLOSE CHANNEL .....	62
6.4.29	RECEIVE DATA .....	63
6.4.30	SEND DATA .....	64
6.4.31	GET CHANNEL STATUS .....	65
6.4.32	SERVICE SEARCH .....	65
6.4.33	GET SERVICE INFORMATION .....	66
6.4.34	DECLARE SERVICE .....	67
6.4.35	SET FRAMES .....	67
6.4.36	GET FRAME STATUS .....	68
6.4.37	RETRIEVE MULTIMEDIA MESSAGE .....	68
6.4.38	SUBMIT MULTIMEDIA MESSAGE .....	69
6.4.39	DISPLAY MULTIMEDIA MESSAGE .....	70
6.4.40	ACTIVATE .....	71
6.4.41	CONTACTLESS STATE CHANGED .....	71
6.4.42	COMMAND CONTAINER .....	71
6.4.43	ENCAPSULATED SESSION CONTROL .....	71
6.5	Common elements in proactive UICC commands .....	72
6.5.1	Command number .....	72
6.5.2	Device identities .....	72
6.5.3	Alpha identifier .....	72
6.5.4	Icon identifiers .....	72
6.5.5	Text Attribute .....	73
6.5.6	Frame identifier .....	73
6.6	Structure of proactive UICC commands .....	73
6.6.1	DISPLAY TEXT .....	73
6.6.2	GET INKEY .....	74
6.6.3	GET INPUT .....	74
6.6.4	MORE TIME .....	75
6.6.5	PLAY TONE .....	75
6.6.6	POLL INTERVAL .....	75
6.6.7	SET-UP MENU .....	76
6.6.8	SELECT ITEM .....	76
6.6.9	SEND SHORT MESSAGE .....	77
6.6.10	Void .....	77
6.6.11	Void .....	77
6.6.12	SET UP CALL .....	77

6.6.13	REFRESH .....	78
6.6.14	POLLING OFF .....	78
6.6.15	PROVIDE LOCAL INFORMATION .....	78
6.6.16	SET UP EVENT LIST .....	79
6.6.17	PERFORM CARD APDU .....	79
6.6.18	POWER OFF CARD .....	79
6.6.19	POWER ON CARD .....	79
6.6.20	GET READER STATUS .....	79
6.6.21	TIMER MANAGEMENT .....	80
6.6.22	SET UP IDLE MODE TEXT .....	80
6.6.23	RUN AT COMMAND .....	80
6.6.24	SEND DTMF COMMAND .....	81
6.6.25	LANGUAGE NOTIFICATION .....	81
6.6.26	LAUNCH BROWSER .....	81
6.6.27	OPEN CHANNEL .....	82
6.6.27.1	OPEN CHANNEL related to CS bearer .....	82
6.6.27.2	OPEN CHANNEL related to packet data service bearer .....	83
6.6.27.3	OPEN CHANNEL related to local bearer .....	84
6.6.27.4	OPEN CHANNEL related to Default (network) Bearer .....	85
6.6.27.5	OPEN CHANNEL related to UICC Server Mode .....	86
6.6.27.6	OPEN CHANNEL related to Terminal Server Mode .....	86
6.6.28	CLOSE CHANNEL .....	86
6.6.29	RECEIVE DATA .....	87
6.6.30	SEND DATA .....	87
6.6.31	GET CHANNEL STATUS .....	87
6.6.32	SERVICE SEARCH .....	87
6.6.33	GET SERVICE INFORMATION .....	88
6.6.34	DECLARE SERVICE .....	88
6.6.35	SET FRAMES .....	88
6.6.36	GET FRAMES STATUS .....	89
6.6.37	RETRIEVE MULTIMEDIA MESSAGE .....	89
6.6.38	SUBMIT MULTIMEDIA MESSAGE .....	90
6.6.39	DISPLAY MULTIMEDIA MESSAGE .....	90
6.6.40	ACTIVATE .....	90
6.6.41	CONTACTLESS STATE CHANGED .....	90
6.6.42	COMMAND CONTAINER .....	91
6.6.43	ENCAPSULATED SESSION CONTROL .....	91
6.7	Command results .....	91
6.8	Structure of TERMINAL RESPONSE .....	93
6.8.1	Command details .....	95
6.8.2	Device identities .....	95
6.8.3	Result .....	96
6.8.4	Duration .....	96
6.8.5	Text string .....	96
6.8.6	Item identifier .....	96
6.8.7	Local information .....	96
6.8.8	Call control requested action .....	97
6.8.9	Result data object 2 .....	97
6.8.10	Card reader status .....	97
6.8.11	Card ATR .....	97
6.8.12	R-APDU .....	98
6.8.13	Timer identifier .....	98
6.8.14	Timer value .....	98
6.8.15	AT Response .....	98
6.8.16	Text string 2 .....	98
6.8.17	Channel data .....	98
6.8.18	Channel status .....	98
6.8.19	Channel data length .....	98
6.8.20	Bearer description .....	98
6.8.21	Buffer size .....	99
6.8.22	Total display duration .....	99
6.8.23	Service Availability .....	99

6.8.24	Service Record.....	99
6.8.25	Other address (local address).....	99
6.8.26	Frames Information.....	99
6.8.27	SA template .....	99
6.8.28	eCAT sequence number.....	100
6.8.29	Encrypted TLV list .....	100
6.8.30	MAC .....	100
6.9	Proactive UICC session and terminal display interaction .....	100
6.10	Handling of unknown, unforeseen and erroneous messages .....	100
6.10.1	General.....	100
6.10.2	Message too short .....	101
6.10.3	Missing minimum information .....	101
6.10.4	Unknown Tag value.....	101
6.10.5	Unexpected Tag value .....	101
6.10.6	Length errors.....	101
6.10.7	Contents not understood .....	102
6.10.8	Extended length data objects .....	102
6.11	Proactive commands versus possible terminal response .....	102
7	ENVELOPE commands .....	107
7.1	Void.....	107
7.2	Menu selection .....	107
7.2.1	Procedure .....	107
7.2.2	Structure of ENVELOPE (MENU SELECTION).....	107
7.3	Call Control by NAA .....	108
7.3.1	Call Control by NAA.....	108
7.3.1.1	Procedure for mobile originated calls .....	108
7.3.1.2	Void.....	108
7.3.1.3	Indication to be given to the user .....	108
7.3.1.4	Interaction with Fixed Dialling Number (FDN).....	109
7.3.1.5	Support of Barred Dialling Number (BDN) service.....	110
7.3.1.6	Structure of ENVELOPE (CALL CONTROL) .....	110
7.4	Timer expiration .....	112
7.4.1	Description.....	112
7.4.2	Structure of ENVELOPE (TIMER EXPIRATION).....	112
7.5	Event download.....	113
7.5.1	MT call event.....	113
7.5.1.1	Procedure .....	113
7.5.1.2	Structure of ENVELOPE (EVENT DOWNLOAD - MT call) .....	113
7.5.2	Call connected event .....	114
7.5.2.1	Procedure .....	114
7.5.2.2	Structure of ENVELOPE (EVENT DOWNLOAD - call connected).....	114
7.5.3	Call disconnected event .....	115
7.5.3.1	Procedure .....	115
7.5.3.2	Structure of ENVELOPE (EVENT DOWNLOAD - call disconnected) .....	115
7.5.4	Location status event .....	116
7.5.4.1	Procedure .....	116
7.5.4.2	Structure of ENVELOPE (EVENT DOWNLOAD - Location status).....	116
7.5.5	User activity event .....	117
7.5.5.1	Procedure .....	117
7.5.5.2	Structure of ENVELOPE (EVENT DOWNLOAD - User activity) .....	117
7.5.6	Idle screen available event.....	117
7.5.6.1	Procedure .....	117
7.5.6.2	Structure of ENVELOPE (EVENT DOWNLOAD - Idle screen available) .....	118
7.5.7	Card reader status event .....	118
7.5.7.1	Procedure .....	118
7.5.7.2	Structure of ENVELOPE (EVENT DOWNLOAD - Card reader status) .....	118
7.5.8	Language selection event.....	119
7.5.8.1	Procedure .....	119
7.5.8.2	Structure of ENVELOPE (EVENT DOWNLOAD - Language selection) .....	119
7.5.9	Browser termination event.....	119
7.5.9.1	Procedure .....	120

7.5.9.2	Structure of ENVELOPE (EVENT DOWNLOAD - Browser termination) .....	120
7.5.10	Data available event .....	120
7.5.10.1	Procedure .....	120
7.5.10.2	Structure of ENVELOPE (EVENT DOWNLOAD - Data available) .....	121
7.5.11	Channel status event .....	121
7.5.11.1	Procedure .....	121
7.5.11.2	Structure of ENVELOPE (EVENT DOWNLOAD - Channel status).....	122
7.5.12	Access Technology Change Event.....	122
7.5.12.1	Procedure .....	122
7.5.12.2	Structure of ENVELOPE (EVENT DOWNLOAD - Access Technology Change).....	123
7.5.13	Display parameters changed event.....	123
7.5.13.1	Procedure .....	123
7.5.13.2	Structure of ENVELOPE (EVENT DOWNLOAD - Display parameters changed).....	124
7.5.14	Local Connection event .....	124
7.5.14.1	Procedure .....	124
7.5.14.2	Structure of ENVELOPE (EVENT DOWNLOAD - Local Connection).....	124
7.5.15	Network Search Mode Change Event.....	125
7.5.15.1	Procedure .....	125
7.5.15.2	Structure of ENVELOPE (EVENT DOWNLOAD - Network Search Mode Change).....	125
7.5.16	Browsing status event .....	126
7.5.16.1	Procedure .....	126
7.5.16.2	Structure of ENVELOPE (EVENT DOWNLOAD - Browsing status) .....	126
7.5.17	Frames Information changed event.....	126
7.5.17.1	Procedure .....	126
7.5.17.2	Structure of ENVELOPE (EVENT DOWNLOAD - Frames Information changed).....	127
7.5.18	HCI connectivity event .....	127
7.5.18.1	Procedure .....	127
7.5.18.2	Structure of ENVELOPE (EVENT DOWNLOAD - HCI connectivity event).....	127
7.5.19	Contactless state request .....	128
7.5.19.1	Procedure .....	128
7.5.19.2	Structure of ENVELOPE(EVENT DOWNLOAD - Contactless state request).....	128
7.5.20	Profile Container.....	128
7.5.20.1	Procedure .....	129
7.5.20.2	Structure of ENVELOPE (EVENT DOWNLOAD - Profile Container) .....	129
7.5.21	Void .....	131
7.6	MMS Transfer Status .....	131
7.6.1	Procedure .....	131
7.6.2	Structure of ENVELOPE (MMS Transfer Status).....	131
7.7	MMS notification download.....	132
7.7.1	Procedure .....	132
7.7.2	Structure of ENVELOPE (MMS notification download).....	133
7.8	Terminal Applications .....	133
7.8.1	Description.....	133
7.8.2	Structure of ENVELOPE (TERMINAL APPLICATIONS) .....	134
7.9	Envelope Container .....	134
7.9.1	Description.....	134
7.9.2	Structure of ENVELOPE (ENVELOPE CONTAINER).....	135
7.10	Service List Retrieval .....	136
7.10.1	Description.....	136
7.10.2	Structure of ENVELOPE (SERVICE LIST) .....	136
8	COMPREHENSION-TLV data objects .....	137
8.1	Address.....	137
8.2	Alpha identifier .....	138
8.3	Subaddress.....	138
8.4	Capability configuration parameters .....	139
8.5	Void.....	139
8.6	Command details.....	139
8.7	Device identities .....	145
8.8	Duration.....	145
8.9	Item .....	146
8.10	Item identifier .....	146

8.11	Response length.....	147
8.12	Result.....	147
8.12.1	Void.....	149
8.12.2	Additional information for terminal problem .....	149
8.12.3	Additional information for network problem.....	149
8.12.4	Void.....	149
8.12.5	Void.....	149
8.12.6	Void.....	149
8.12.7	Void.....	149
8.12.8	Additional information for interaction with call control.....	149
8.12.9	Additional information for MultipleCard commands .....	150
8.12.10	Additional information for launch browser problem .....	150
8.12.11	Additional information for Bearer Independent Protocol .....	150
8.12.12	Additional information for Frames commands .....	151
8.12.13	Additional information for SUBMIT and RETRIEVE MULTIMEDIA MESSAGE.....	151
8.13	3GPP- SMS TPDU .....	152
8.14	Void.....	152
8.15	Text string .....	152
8.15.1	Coding of text in unpacked format .....	152
8.15.2	Coding of text in packed format .....	152
8.15.3	Coding of text in 16 bits UCS2 alphabet format.....	152
8.16	Tone.....	153
8.17	Void.....	154
8.18	File list.....	154
8.19	Location information .....	154
8.20	IMEI .....	155
8.21	Help request.....	155
8.22	Network measurement results .....	155
8.23	Default text.....	155
8.24	Items next action indicator .....	155
8.25	Event list.....	156
8.26	Cause .....	157
8.27	Location status.....	157
8.28	Transaction identifier .....	158
8.29	Void.....	158
8.30	Call control requested action .....	158
8.31	Icon identifier .....	158
8.32	Item icon identifier list .....	159
8.33	Card reader status .....	159
8.34	Card ATR.....	160
8.35	C-APDU .....	161
8.36	R-APDU .....	161
8.37	Timer identifier .....	161
8.38	Timer value .....	162
8.39	Date-time and time zone.....	162
8.40	AT command.....	163
8.41	AT response.....	163
8.42	BC repeat indicator.....	163
8.43	Immediate response .....	163
8.44	DTMF string.....	164
8.45	Language .....	164
8.46	Void.....	164
8.47	Browser identity .....	164
8.48	URL.....	165
8.49	Bearer .....	165
8.50	Provisioning file reference .....	165
8.51	Browser termination cause .....	165
8.52	Bearer description.....	166
8.53	Channel data.....	167
8.54	Channel data length .....	167
8.55	Buffer size .....	167
8.56	Channel status .....	167



8.57	Card reader identifier.....	168
8.58	Other Address.....	169
8.59	UICC/terminal interface transport level .....	169
8.60	AID.....	170
8.61	Access technology .....	170
8.62	Display parameters .....	171
8.63	Service record.....	171
8.64	Device filter.....	173
8.65	Service search.....	174
8.66	Attribute information.....	174
8.67	Service availability .....	175
8.68	Remote entity address .....	176
8.69	ESN .....	177
8.70	Network access name .....	177
8.71	CDMA-SMS TPDU .....	177
8.72	Text attribute .....	177
8.73	Item text attribute list .....	177
8.74	IMEISV .....	178
8.75	Network search mode.....	178
8.76	Battery State .....	179
8.77	Browsing status .....	179
8.78	Frame Layout .....	179
8.79	Frames Information .....	180
8.80	Frame identifier .....	181
8.81	MEID.....	181
8.82	Multimedia Message Reference .....	181
8.83	Multimedia Message Identifier.....	181
8.84	Multimedia Message Transfer status .....	182
8.85	MM Content Identifier .....	182
8.86	Multimedia Message Notification .....	182
8.87	Last Envelope.....	182
8.88	Registry application data .....	183
8.89	Activate descriptor .....	184
8.90	Broadcast Network information .....	184
8.91	Contactless state request.....	185
8.92	Contactless functionality state .....	186
8.93	Extended registry application data .....	186
8.94	eCAT client profile.....	187
8.95	eCAT client identity .....	187
8.96	Encapsulated envelope type .....	187
8.97	Call control result .....	187
8.98	eCAT sequence number .....	188
8.99	Encrypted TLV list.....	188
8.100	MAC.....	189
8.101	SA template .....	189
8.102	CAT service list.....	191
9	Tag values .....	192
9.1	BER-TLV tags in terminal to UICC direction.....	192
9.2	BER-TLV tags in UICC to terminal direction.....	193
9.3	COMPREHENSION-TLV tags in both directions.....	193
9.4	Type of command and next action indicator .....	193
10	Allowed type of command and device identity combinations .....	194
11	Void.....	195
<b>Annex A (normative):</b>	<b>Support of CAT by terminal equipment.....</b>	<b>196</b>
<b>Annex B (informative):</b>	<b>Example of DISPLAY TEXT proactive UICC command.....</b>	<b>198</b>
<b>Annex C (normative):</b>	<b>Structure of CAT communications .....</b>	<b>199</b>

<b>Annex D (informative):</b>	<b>Terminal display in proactive UICC session.....</b>	<b>200</b>
<b>Annex E (informative):</b>	<b>Help information feature processing.....</b>	<b>201</b>
<b>Annex F (informative):</b>	<b>Monitoring of events.....</b>	<b>202</b>
<b>Annex G (normative):</b>	<b>Support of multiple card operation.....</b>	<b>203</b>
<b>Annex H (informative):</b>	<b>Multiple card proactive command examples .....</b>	<b>204</b>
<b>Annex I (informative):</b>	<b>Bearer independent protocol proactive command examples .....</b>	<b>205</b>
<b>Annex J (informative):</b>	<b>WAP terminology .....</b>	<b>210</b>
<b>Annex K (informative):</b>	<b>Use of CAT bearer independent protocol for local links Bluetooth case .....</b>	<b>211</b>
K.1	Service search command.....	211
K.2	Get service information command .....	211
K.3	OPEN CHANNEL command.....	211
<b>Annex L (informative):</b>	<b>Bluetooth service discovery protocol.....</b>	<b>214</b>
L.1	Service attribute.....	215
<b>Annex M (informative):</b>	<b>Use of CAT bearer independent protocol for local links, server case .....</b>	<b>216</b>
<b>Annex N (informative):</b>	<b>Browsing terminology.....</b>	<b>217</b>
<b>Annex O (informative):</b>	<b>Use of USAT Bearer independent protocol for local links IrDA case.....</b>	<b>218</b>
O.1	Service Search command .....	218
O.2	Get Service Information command .....	218
O.3	OPEN CHANNEL command.....	219
<b>Annex P (informative):</b>	<b>IrDA Information Access Service.....</b>	<b>221</b>
<b>Annex Q (informative):</b>	<b>IrDA IAS class name and associated parameters .....</b>	<b>222</b>
<b>Annex R (informative):</b>	<b>Example of Frames usage.....</b>	<b>223</b>
<b>Annex S (normative):</b>	<b>Support of CAT by Terminals with reduced feature capabilities. ....</b>	<b>226</b>
<b>Annex T (normative):</b>	<b>Default routing for CAT over the modem interface .....</b>	<b>229</b>
T.1	Default routing mechanism .....	230
T.2	Combination rules for terminal profiles .....	231
<b>Annex U (informative):</b>	<b>Change history .....</b>	<b>232</b>
History	.....	236

---

## Intellectual Property Rights

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

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

---

## Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee Smart Card Platform (SCP).

It is based on work originally done in the 3GPP in TSG-terminals WG3.

The contents of the present document are subject to continuing work within TC SCP and may change following formal TC SCP approval. If TC SCP modifies the contents of the present document, it will then be republished by ETSI with an identifying change of release date and an increase in version number as follows:

Version x.y.z

where:

- x the first digit:
  - 0 early working draft;
  - 1 presented to TC SCP for information;
  - 2 presented to TC SCP for approval;
  - 3 or greater indicates TC SCP approved document under change control.
- Y the second digit is incremented for all changes of substance, i.e. technical enhancements, corrections, updates, etc.
- z the third digit is incremented when editorial only changes have been incorporated in the document.

---

## Modal verbs terminology

In the present document "**shall**", "**shall not**", "**should**", "**should not**", "**may**", "**may not**", "**need**", "**need not**", "**will**", "**will not**", "**can**" and "**cannot**" are to be interpreted as described in clause 3.2 of the [ETSI Drafting Rules](#) (Verbal forms for the expression of provisions).

"**must**" and "**must not**" are **NOT** allowed in ETSI deliverables except when used in direct citation.

---

# 1 Scope

The present document defines the interface between the UICC and the terminal, and mandatory terminal procedures, specifically for "NAA Card Application Toolkit".

The Card Application Toolkit (CAT) is a set of generic commands and procedures for use by the ICC, irrespective of the access technology of the network. Within the scope of the present document, the UICC refers here to an ICC which supports at least one application in order to access a network. This application is called here Network Access Application (NAA).

The ICC is considered as a platform, which is either based on TS 102 221 [1] or TS 102 600 [38], here called "3G platform", or TS 151 011 [8], here called "2G platform".

NAA can be:

- a USIM application, as defined in TS 131 102 [6], which can reside only on a 3G platform;
- a SIM application, as defined in TS 151 011 [8], which can reside either on a 3G or a 2G platform;
- a TSIM application, as defined in TS 100 812 [i.2], which can reside only on a 3G platform;
- a ISIM application, as defined in TS 131 103 [36], which can reside only on a 3G platform;
- a RUIM application, as defined in TIA/IS-820-A [17], 3GPP2 C.S0023-0 [30], which can reside on a 2G platform; or
- other applications residing on a 3G platform or a 2G platform. Specifying the interface is to ensure interoperability between an ICC and a terminal independently of the respective manufacturers and operators.

The present document specifies as well mechanisms in order to expand the generic set of commands and procedures by access technology specific ones.

The present document defines:

- the commands;
- the application protocol;
- the mandatory requirements on the ICC and terminal for each procedure.

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

---

# 2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the reference document (including any amendments) applies.

- In the case of a reference to a TC SCP document, a non specific reference implicitly refers to the latest version of that document in the same Release as the present document.

Referenced documents which are not found to be publicly available in the expected location might be found at <http://docbox.etsi.org/Reference>.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

## 2.1 Normative references

The following referenced documents are necessary for the application of the present document.

- [1] ETSI TS 102 221: "Smart Cards; UICC-Terminal interface; Physical and logical characteristics".
- [2] ETSI TS 122 001: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Principles of circuit telecommunication services supported by a Public Land Mobile Network (PLMN) (3GPP TS 22.001)".
- [3] ETSI TS 123 038: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Alphabets and language-specific information (3GPP TS 23.038)".
- [4] Void.
- [5] ETSI TS 127 007: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; AT command set for User Equipment (UE) (3GPP TS 27.007)".
- [6] ETSI TS 131 102: "Universal Mobile Telecommunications System (UMTS); LTE; Characteristics of the Universal Subscriber Identity Module (USIM) application (3GPP TS 31.102)".
- [7] ETSI TS 131 110: "Universal Mobile Telecommunications System (UMTS); Numbering system for telecommunication IC card applications (3GPP TS 31.110)".
- [8] ETSI TS 151 011: "Digital cellular telecommunications system (Phase 2+); Specification of the Subscriber Identity Module - Mobile Equipment (SIM-ME) interface (3GPP TS 51.011)".
- [9] IETF RFC 768: "User Datagram Protocol".
- [10] IETF RFC 793: "Transmission Control Protocol".
- [11] IETF RFC 1738: "Uniform Resource Locators (URL)".

NOTE: Obsoleted by RFC 4248 and RFC 4266.

- [12] ISO 639 (all parts): "Codes for the representation of names of languages".
- [13] ISO/IEC 7816-3: "Identification cards -- Integrated circuit cards -- Part 3: Cards with contacts -- Electrical interface and transmission protocols".
- [14] ISO/IEC 7816-4: "Identification cards -- Integrated circuit cards -- Part 4: Organization, security and commands for interchange".
- [15] Void.
- [16] Specification of the Bluetooth system; Volume 2; Profiles of the Bluetooth system.

NOTE: Available at <http://www.bluetooth.org/>.

- [17] TIA/EIA/IS-820-A: "Removable User Identity Module (R-UIM) for TIA/EIA Spread Spectrum Standards".
- [18] ANSI/TIA/EIA-41 (December 1997): "Cellular Radiotelecommunications Intersystem Operations".
- [19] ETSI TS 100 922: "Digital cellular telecommunications system (Phase 2+) (GSM); Subscriber Identity Modules (SIM); Functional characteristics (GSM 02.17)".
- [20] ETSI TS 124 008: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Mobile radio interface Layer 3 specification; Core network protocols; Stage 3 (3GPP TS 24.008)".
- [21] Void.

- [22] Recommendation ITU-T E.164: "The international public telecommunication numbering plan".
- [23] Recommendation ITU-T X.121: "International numbering plan for public data networks".
- [24] Recommendation ITU-T F.69: "The international telex service - Service and operational provisions of telex destination codes and telex network identification codes".
- [25] ANSI/TIA/EIA-136-C: "TDMA Third Generation Wireless".
- [26] ETSI TS 131 111: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Universal Subscriber Identity Module (USIM) Application Toolkit (USAT) (3GPP TS 31.111)".
- [27] ETSI TS 123 040: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Technical realization of the Short Message Service (SMS) (3GPP TS 23.040)".
- [28] ETSI TS 122 030: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Man-Machine Interface (MMI) of the User Equipment (UE) (3GPP TS 22.030)".
- [29] 3GPP2 C.S0015-0: "Short Message Service (SMS)".

NOTE: Available at [http://www.3gpp2.org/public\\_html/specs/CS0015-0.pdf](http://www.3gpp2.org/public_html/specs/CS0015-0.pdf).

- [30] 3GPP2 C.S0023-0: "Removable User Identity Module (R-UIM) for cdma2000 Spread Spectrum Systems".

NOTE: Available at [http://www.3gpp2.org/public\\_html/specs/CS0023-0.pdf](http://www.3gpp2.org/public_html/specs/CS0023-0.pdf).

- [31] ETSI TS 101 220: "Smart Cards; ETSI numbering system for telecommunication application providers".
- [32] ETSI TS 123 003: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Numbering, addressing and identification (3GPP TS 23.003)".

- [33] Infrared Data Association: "Serial Infrared Link Management Protocol (IrLMP)", version 1.1.

NOTE: Available at <http://www.irda.org/>.

- [34] Void.

- [35] Void.

- [36] ETSI TS 131 103: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); LTE; Characteristics of the IP Multimedia Services Identity Module (ISIM) application (3GPP TS 31.103)".

- [37] ETSI TS 123 140: "Digital cellular telecommunications system (Phase 2+); Universal Mobile Telecommunications System (UMTS); Multimedia Messaging Service (MMS); Functional description; Stage 2 (3GPP TS 23.140)".

- [38] ETSI TS 102 600: "Smart Cards; UICC-Terminal interface; Characteristics of the USB interface".

- [39] ETSI TS 102 613: "Smart Cards; UICC - Contactless Front-end (CLF) Interface; Part 1: Physical and data link layer characteristics".

- [40] ETSI TS 102 622: "Smart Cards; UICC - Contactless Front-end (CLF) Interface; Host Controller Interface (HCI)".

- [41] OMA Device Management V1.2.

- [42] OMA Data Synchronization V1.2.1.

NOTE: Available at [http://www.openmobilealliance.org/technical/release\\_program/ds\\_v12.aspx](http://www.openmobilealliance.org/technical/release_program/ds_v12.aspx).

- [43] ETSI EN 300 468: "Digital Video Broadcasting (DVB); Specification for Service Information (SI) in DVB systems".
- [44] ETSI EN 302 304: "Digital Video Broadcasting (DVB); Transmission System for Handheld Terminals (DVB-H)".
- [45] ETSI TS 102 589: "Forward Link Only Air Interface; Specification for Terrestrial Mobile; Multimedia Multicast".
- [46] IEEE 802.16-2009: "Local and Metropolitan Area Networks - Part 16: Air Interface for Broadband Wireless Access Systems".
- [47] 3GPP2 C.S0035-A v3.0: "CDMA Card Application Toolkit (CCAT)".
- NOTE: Available at [http://www.3gpp2.org/public\\_html/specs/C.S0035-A%20v3.0%20CCAT\\_20120828.pdf](http://www.3gpp2.org/public_html/specs/C.S0035-A%20v3.0%20CCAT_20120828.pdf).
- [48] ETSI TS 102 484: "Smart Cards; Secure channel between a UICC and an end-point terminal".
- [49] TIA/EIA-553: "Mobile Station - Base Station Compatibility Standard".
- [50] TIA/EIA-95: "Mobile Station-Base Station Compatibility Standard for Wideband Spread Spectrum Cellular Systems".
- [51] TIA/EIA/IS-2000: "Physical Layer Standard for cdma2000® Spread Spectrum Systems".
- [52] TIA/EIA/IS-856: "cdma2000® High Rate Packet Data Air Interface Specification".
- [53] 3GPP2 C.S0065-B: "cdma2000 Application on UICC for Spread Spectrum Systems".

## 2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

- [i.1] ETSI TS 100 906: "Digital cellular telecommunications system (Phase 2+) (GSM); Mobile Stations (MS) features (GSM 02.07)".
- [i.2] ETSI TS 100 812 (all parts): "Terrestrial Trunked Radio (TETRA); Subscriber Identity Module to Mobile Equipment (TSIM-ME) interface".
- [i.3] Recommendation ITU-T E.163: "Numbering plan for the international telephone service".

---

# 3 Definitions, symbols and abbreviations

## 3.1 Definitions

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

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

**application protocol:** set of procedures required by the application

**bearer independent protocol:** mechanism by which the terminal provides the UICC with access to the data bearers supported by the terminal and the network

**Card Application Toolkit (CAT):** set of applications and related procedures that may be used during a card session

**card reader x:** electrical interface compatible with ISO/IEC 7816-3 [13] to support additional card

**card session:** link between the card and the external world, using APDUs, starting with the ATR and ending with a subsequent reset or a deactivation of the card

**NOTE:** A card session may take place either over the electrical interface specified in TS 102 221 [1] or over the Smart Card functional interface specified in TS 102 600 [38].

**Card x:** additional card using an interface according to ISO/IEC 7816-3 [13]

**CAT client:** component in the Connected Entity providing a subset of the Connected Entity's CAT facilities

**connected entity:** logical entity (consisting of any combination of hardware and/or software) that can support CAT facilities via a modem interface, e.g. using AT commands

**data channel:** allow the UICC and the network to exchange data using a selected bearer

**data object:** information seen at the interface for which are defined a tag (identifier), a length and a value

**NOTE:** Data objects can be either BER-TLV or COMPREHENSION-TLV as defined in TS 101 220 [31]. In the present document, all BER-TLV data objects are "primitive": the value part consists only of COMPREHENSION-TLV data objects. Unless otherwise noted, a reference to a TLV is to a BER-TLV.

**eCAT:** mechanisms of encapsulated profiles, commands, responses and envelopes used in the communication between a UICC and eCAT clients ("encapsulated CAT")

**eCAT client:** entity within the terminal that is able to provide CAT facilities and processes encapsulated commands and events: COMMAND CONTAINERS, ENCAPSULATED SESSION CONTROL commands and associated TERMINAL RESPONSEs and issues Encapsulated Profile and Envelope Container events

**encapsulated command session:** communication sequence between an eCAT client and the UICC, starting with the assignment of a client identity by the UICC and ending upon request by either entity or with the end of the card session

**link:** radio resource

**modem:** component in the terminal that provides interfaces to the mobile network, to the UICC and to a Connected Entity

**NOTE:** The modem may have a limited set of capabilities.

**Multi-Media Call:** services that handle several types of media such as audio and video in a synchronized way from the user's point of view

**network access application:** application residing in the UICC which holds a subscriber identity and an authentication algorithm and provides the access to a network

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

**proactive UICC:** UICC which is capable of issuing commands to the terminal

**proactive UICC session:** sequence of related CAT commands and responses which starts with the status response '91XX' (proactive command pending) and ends with a status response of '90 00' (normal ending of command) after Terminal Response

**Rx buffer:** dedicated memory used to temporarily store data to be retrieved

**Service Data Unit (SDU):** set of data in layered systems that is sent by a user of the services of a given layer, and is transmitted to a peer service user semantically unchanged

**NOTE:** A Protocol Control Information (PCI) header is attached to the Service Data Unit (SDU) by the layer to form a Protocol Data Unit (PDU).

**Tx buffer:** dedicated memory used to temporarily store data to be sent

**UICC:** smart card that conforms to the specification written and maintained by the ETSI Smart Card Platform project

**NOTE:** UICC is neither an abbreviation nor an acronym.



**UICC application session:** execution of a sequence of commands internal to the UICC that can result in the performance of one or several proactive UICC sessions

NOTE: The UICC application session can be started by any event in the card session, and can execute for the duration of the card session. Processing of the UICC application session will not interfere with normal 3G operation.

## 3.2 Symbols

For the purposes of the present document, the following symbol applies:

"0" to "9" and "A" to "F" The sixteen hexadecimal digits

## 3.3 Abbreviations

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

APDU	Application Protocol Data Unit
ATR	Answer To Reset
BC	Bearer Capability
BCCCH	Broadcast Control Channel
BCD	Binary Coded Decimal
BD_ADDR	Bluetooth Device ADDRESS
BDN	Barred Dialling Number
BER	Basic Encoding Rules of ASN.1
BIP	Bearer Independent Protocol
BSID	Base Station Identifier
C-APDU	Command Application Protocol Data Unit
CAT	Card Application Toolkit
CHTML	Compact HyperText Markup Language
CoD	Class of Device (Bluetooth related)
CSD	Circuit Switched Data
CSG	Closed Subscriber Group
CSIM	cdma2000 Subscriber Identity Module
DCS	Data Coding Scheme
DTMF	Dual Tone Multiple Frequency
DVB-H	Digital Video Broadcasting - Handheld
DVB-SH	Digital Video Broadcasting - Satellite services to Handhelds
DVB-T	Digital Video Broadcasting - Terrestrial
EF	Elementary File
EIA	Electronics Industries Association
ESN	Electronic Serial Number
FDN	Fixed Dialling Number
FFS	For Further Study
FLO	Forward-Link Only
GSM	Global System for Mobile communications
HRPD	High Rate Packet Data
HTML	HyperText Markup Language
HTTP	HyperText Transfer Protocol
IAS	Information Access Service (IrDA related)
ID	Identifier
IEC	International Electrotechnical Commission
IMEI	International Mobile Equipment Identity
IMEISV	International Mobile Equipment Identity and Software Version
IMSI	International Mobile Subscriber Identity
IP	Internet Protocol
Ipv4	Internet Protocol version 4
IrLMP	Infrared Link Management Protocol (IrDA related)
ISO	International Organization for Standardization
LAC	Location Area Code

Igth	the (specific) length of a data unit
LSAP	Link Service Access Point (IrDA related)
MEID	Mobile Equipment Identifier
MM	Multimedia Message
MMI	Man Machine Interface
MMS	Multimedia Messaging Service
MO	Mobile Originated
MT	Mobile Terminated
NAA	Network Access Application
NAI	Next Action Indicator
NMR	Network Measurement Results
NPI	Numbering Plan Identifier
PC	Personal Computer
PDN	Packet Data Network
PDP	Packet Data Protocol, e.g. IP or X25 or PPP
PDU	Protocol Data Unit
PPP	Point-to-Point Protocol
R-APDU	Response Application Protocol Data Unit
RF	Radio Frequency
RFCOMM	Radio Frequency COMMunication
RFM	Remote File Management
RFU	Reserved for Future Use
RUIM	Removable User Identity Module
SDP	Service Discovery Protocol (Bluetooth related)
SDU	Service Data Unit
SGSN	Serving GPRS Support Node
SMS	Short Message Service
SMSC	Short Message Service Centre
SW1/SW2	Status Word 1/Status Word 2
TCP	Transmission Control Protocol
T-DMB	Terrestrial - Digital Multimedia Broadcasting
TE	Terminal Equipment (e.g. an attached personal computer)
TETRA	Terrestrial Trunked Radio
TIA	Telecommunications Industries Association
TLV	Tag, Length, Value
TON	Type Of Number
TP	Transfer layer Protocol
TR	TERMINAL RESPONSE
TSIM	TETRA SIM application
TTP	Tiny Transport Protocol
UCS2	Universal two byte coded Character Set
UDP	User Datagram Protocol
UE	User Equipment
UMTS	Universal Mobile Telecommunication System
URL	Uniform Resource Location
USAT	USIM Application Toolkit
USSD	Unstructured Supplementary Service Data
UTRAN	UMTS Terrestrial Radio Access Network
UUID	Universally Unique Identifier
WAE	Wireless Application Environment
WAP	Wireless Application Protocol
WiMAX	Worldwide Interoperability for Microwave Access
WML	Wireless Markup Language
XHTML	eXtensible HyperText Markup Language

---

## 4 Overview of CAT

The CAT provides mechanisms which allow applications, existing in the UICC, to interact and operate with any terminal which supports the specific mechanism(s) required by the application.

If class "a" is supported, a UICC supporting CAT shall be able to communicate with the additional card(s) and get information about the additional reader(s) via the terminal.

The following mechanisms have been defined. These mechanisms are dependent upon the commands and protocols relevant to CAT as USAT in TS 102 221 [1] for a 3G platform and as SAT in TS 151 011 [8] for a 2G platform.

### 4.1 Profile download

Profile downloading provides a mechanism for the terminal to tell the UICC what it is capable of.

### 4.2 Proactive UICC

Proactive UICC gives a mechanism whereby the UICC can initiate actions to be taken by the terminal. These actions include:

- displaying text from the UICC to the terminal;
- sending a short message;
- setting up a voice call to a number held by the UICC;
- setting up a data call to a number and bearer capabilities held by the UICC;
- playing tone in earpiece;
- initiating a dialogue with the user;
- NAA network access application initialization request and notification of changes to EF(s);
- providing local information from the terminal to the UICC;
- communicating with the additional card(s) (if class "a" is supported);
- providing information about the additional card reader(s) (if class "a" is supported);
- managing timers running physically in the terminal;
- running an AT command received from the UICC, and returning the result to the UICC (if class "b" is supported);
- sending DTMF;
- requesting the terminal to launch the browser corresponding to a URL (if class "c" is supported);
- establishing and managing a bearer independent protocol (if class "e" is supported);
- dividing the terminal's screen into several rectangular regions (frames) (if class "i" is supported);
- requesting the terminal to start an application on the terminal, if this application is registered for such a request (if class "k" is supported);
- activate an interface (if class "l" is supported);
- requesting the terminal to report geographical location information to the UICC (if class "n" is supported);
- providing CAT facilities by a modem and a Connected Entity (if class "s" is supported);

- encapsulating commands for an eCAT client and sending encapsulated profiles and envelopes by an eCAT client (if class "u" is supported);
- securing the data exchanged with an eCAT client (if class "x" is supported).

For each command involved in the dialog with the user, a help information may be available, either for each item of a list of items proposed to the user, or with each command requesting a response from the user. If a proactive command involved in the dialog with the user indicates the availability of the help feature, the support of this feature is optional for the terminal.

### 4.3 Data download to UICC

Data downloading to the UICC uses either dedicated commands (using the transport mechanisms of the technology) or the Bearer independent protocol. Transferral of information over the UICC-terminal interface uses the ENVELOPE command.

### 4.4 Menu selection

A set of possible menu entries is supplied by the UICC in a proactive UICC command. The menu selection mechanism is used to transfer the UICC application menu item which has been selected by the user to the UICC. The menu selection mechanism may also be used for requesting help information on the items of the UICC application menu.

### 4.5 Call control by network access application

When this service is activated by the NAA, all dialled digit strings are first passed to the UICC before the terminal sets up the call. The terminal shall also pass to the UICC at the same time its current serving cell. The toolkit application has the ability to allow, bar or modify the call. The application also has the ability to replace a call request by another call request.

NOTE: In some technologies, the call request can even be replaced by another operation, for instance USSD or SMS in GSM/3GPP.

### 4.6 Void

### 4.7 Event download

A set of events to monitor for is supplied by the UICC in a proactive UICC command. The event download mechanism is used to transfer details of the event to the UICC, when it occurs. Events that the terminal can report to the UICC include incoming calls, location status, access technology, display parameters changed, and availability of the screen for applications.

### 4.8 Security

Applications designed using the features in the present document may require methods to ensure data confidentiality, data integrity, and data sender validation, or any subset of these. Requirements for these mechanisms are defined in clause 11.

### 4.9 Multiple card

This clause applies if class "a" is supported.

One event and a set of proactive commands are supplied to monitor and control Card x behaviour.

## 4.10 Timer expiration

The UICC is able to manage timers running physically in the terminal with a proactive command. The timer expiration mechanism is used to inform the UICC when a timer expires.

## 4.11 Bearer Independent Protocol

The following clause applies if class "e" is supported.

The set of proactive commands (OPEN CHANNEL, CLOSE CHANNEL, SEND DATA, RECEIVE DATA, and GET CHANNEL STATUS) and events (Data available, Channel status) allows the UICC to establish a data channel with the terminal, and through the terminal either to a remote Server in the Network or to a remote device in the Personal Area Network. The UICC provide information for the terminal to select an available bearer at the time of channel establishment. The terminal then allows the UICC and the Server to exchange data on this channel, transparently. The UICC uses service of terminal lower layer to send data by providing Service Data Unit to terminal. The default lower layer is the higher layer of selected bearer.

The following clauses apply if class "f" is supported.

The proactive command SERVICE SEARCH allows the UICC to look for services available on remote devices. The proactive command GET SERVICE INFORMATION allows the UICC to get detailed information regarding one service.

The proactive command DECLARE SERVICE allows the UICC to add or delete a service to the terminal service database. The event Local Connection allows to inform the UICC of a connection request on a local bearer.

## 4.12 Description of the access technology indicator mechanism

This clause describes the mechanisms that can be employed to indicate access technology specific dependencies in a multi-access technology environment.

There are cases where toolkit applications need to know which access technology the terminal is currently in so that it can issue access technology dependent commands as well as determine that the response to a particular command is technology dependent. Setting up the event, ACCESS TECHNOLOGY CHANGE, and its continuous monitoring, provides a means by which the terminal can inform toolkit applications of a change in the current access technology. This change is notified to the toolkit applications via the ENVELOPE command: EVENT DOWNLOAD - "Access Technology Change" together with the new access technology (if single access technology is set up in the event list) or with the list of current access technologies (if multiple access technologies is set up in the event list).

Additionally, the proactive command, PROVIDE LOCAL INFORMATION, can be used to provide an access technology indication. This is achieved by the toolkit application using the Access Technology command qualifier in the PROVIDE LOCAL INFORMATION command to which the terminal responds with the current access technology or access technologies using the TERMINAL RESPONSE message.

In a multi-access technology environment there are some services that are access technology specific (e.g. the SEND USSD proactive command is used in GSM/UTRAN only). In such cases, if the toolkit application issues such a proactive command then the permanent result, "Access Technology unable to process command" is used by the terminal to inform the toolkit application that the requested command could not be performed due to access technology dependencies. Here the toolkit application should not re-issue the command whilst within the same access technology, as the result will be the same, however, it may re-issue the command when in another access technology.

## 4.13 Tag allocation guidelines

The tag allocation guidelines to be followed when requesting a new tag value are described in TS 101 220 [31].

## 4.14 Description of the network search mode mechanism

This clause describes the mechanisms that can be employed to indicate the network search mode.

There are cases where toolkit applications need to know which Network Search Mode is selected by the user so it can issue specific roaming behaviour.

An application for roaming management can be deactivated when a user selects manual mode. When automatic mode is restored the application can be activated again.

Setting up the event, NETWORK SEARCH MODE CHANGE, and its continuous monitoring, provides a means by which the terminal can inform the toolkit application of a change in the current network search mode. This change is notified to the toolkit application via the ENVELOPE command: EVENT DOWNLOAD - "Network Search Mode change" together with the new search mode.

Additionally, the proactive command, PROVIDE LOCAL INFORMATION, can be used to provide a search mode indication. This is achieved by the toolkit application using the Network Search Mode command qualifier in the PROVIDE LOCAL INFORMATION command to which the terminal responds with the current search mode using the TERMINAL RESPONSE message.

## 4.15 CAT operation in reduced capability terminals

The present document takes into account terminal types corresponding to the following reduced capabilities:

- no display capability;
- no keypad available;
- no audio alerting capability;
- no speech call capability;
- no support of multiple languages.

These terminal types are used to identify which CAT features are not available for each type of reduced functionality.

NOTE: Terminal types details are in annex S.

## 4.16 CAT over the modem interface

This clause applies if class "s" is supported.

If a UICC connected to a modem (e.g. a USB modem providing access to mobile networks for a PC) uses CAT, some CAT facilities can be provided by the modem itself, whereas other facilities can be implemented in the entity connected to the modem (Connected Entity). A mechanism is defined in TS 127 007 [5] to transport the CAT commands to the Connected Entity and responses and events from the Connected Entity to the modem and to the UICC over the modem interface using AT commands. The handling of the profiles of the modem and the Connected Entity are also defined in the present document.

In case proactive commands do not explicitly indicate routing information, the default routing mechanism is defined in annex T. These provisions for default routing also apply in a similar way if several CAT clients within the Connected Entity provide CAT facilities and the Connected Entity merges these facilities into one Connected Entity profile.

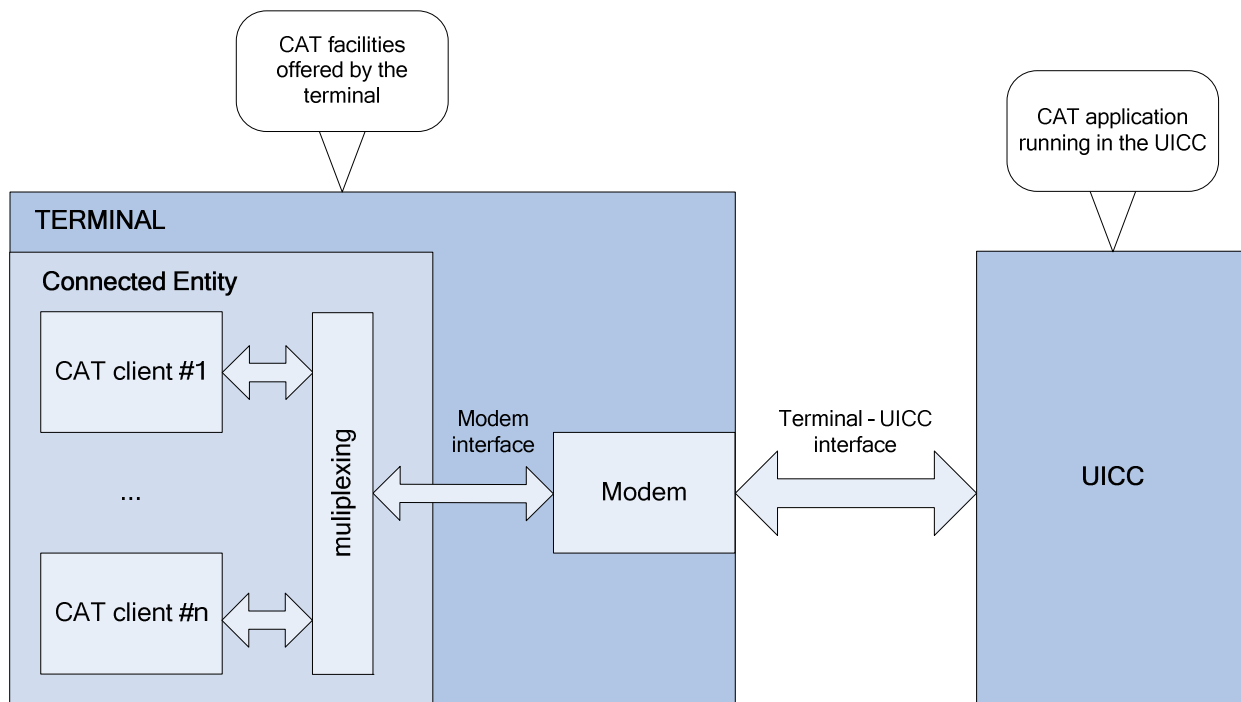


Figure 4.1

## 4.17 CAT facilities provided by eCAT clients

This clause applies if class "u" is supported.

eCAT clients provide a set of CAT facilities of their own, which may overlap with the CAT facilities of other eCAT clients or with the facilities of the terminal. This is enabled by the encapsulation of CAT commands, terminal responses and envelopes in a set of special commands:

- the Profile Container event, to inform the UICC about the profile of an eCAT client and to inform the eCAT client about services provided by the UICC in the encapsulated session;
- COMMAND CONTAINER and associated TERMINAL RESPONSE to embed the command to and response from an eCAT client;
- the envelope container event that allows an eCAT client to send envelopes to the UICC; and
- ENCAPSULATED SESSION CONTROL to end a session with an eCAT client;
- if class "x" is supported in addition, these command may optionally be authenticated or authenticated and encrypted.

As a result of the encapsulation, some data objects appear twice in commands, responses and envelopes: once for the outer structure, the "container" and once for the inner structure, the encapsulated element.

To distinguish between these two layers, the outer structure is described in the present document as being processed by the terminal, whereas the inner structure is described as being processed by the eCAT client. The task assigned logically to the terminal includes wrapping and unwrapping of data provided to/from the eCAT client and the associated routing.

NOTE: This logic does not imply any specific way of implementing this feature in the terminal or in the UICC.

eCAT clients may be part of CAT clients as defined above for CAT over the modem interface or may be part of the terminal using a different (internal) interface.

The usage of keys and the handling of an eCAT secure channel are specified in TS 102 484 [48].

## 5 Profile download

### 5.1 Procedure

The profile download instruction is sent by the terminal to the UICC as part of the UICC initialization procedure and as soon as possible when CAT functionality is modified in the terminal. If class "s" is supported, the profile download instruction is sent also every time the Connected Entity accessing CAT functionalities over a Modem interface is connected or disconnected or changes its profile. If the terminal supports class "s" the profile download instruction shall combine capabilities supported by the terminal and the Connected Entity according to annex T.

This procedure is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform. The profile sent by the terminal shall state the facilities relevant to CAT that are supported by the terminal.

This procedure is important, as it allows the UICC to determine what the terminal is capable of, and the UICC can then limit its instruction range accordingly. If no command is sent by the terminal, the UICC shall assume that the terminal does not support CAT.

### 5.2 Structure and coding of TERMINAL PROFILE

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

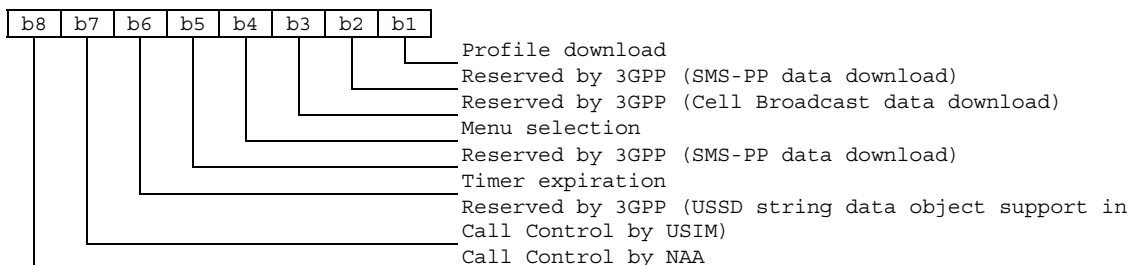
Command parameters/data:

Description	Clause	M/O/C	Length
Profile	-	M	lgth

Profile:

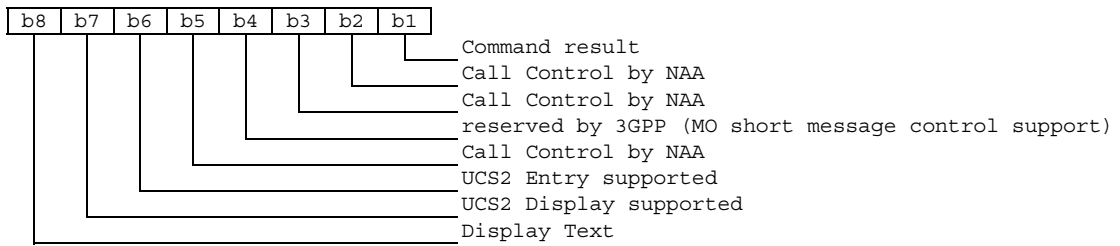
- Contents:
  - The list of CAT facilities that are supported by the terminal.
- Coding:
  - 1 bit is used to code each facility:
    - bit = 1: facility supported by terminal;
    - bit = 0: facility not supported by terminal.

First byte (Download):

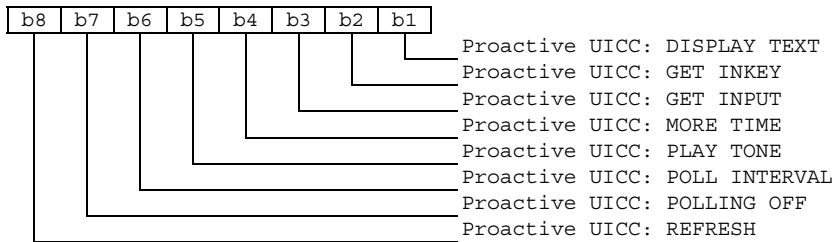




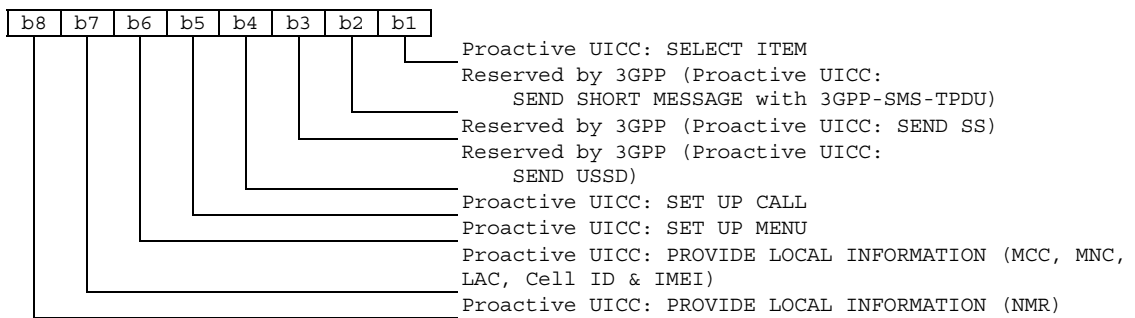
Second byte (Other):



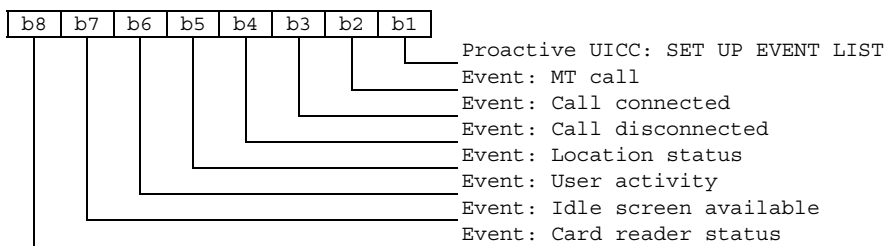
Third byte (Proactive UICC):



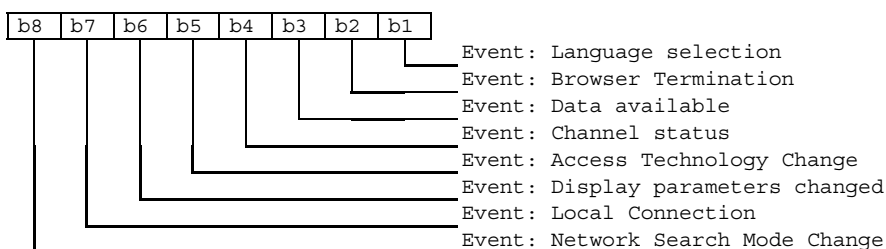
Fourth byte (Proactive UICC):



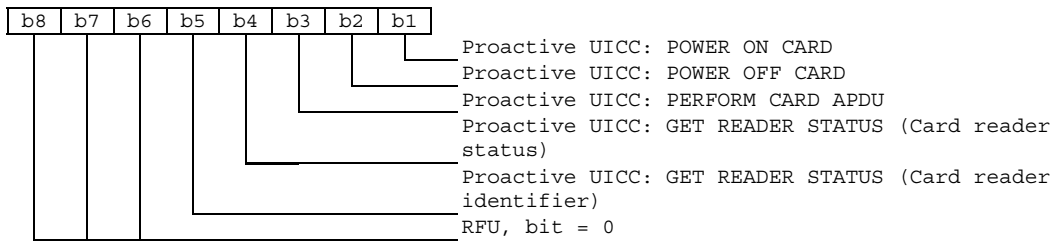
Fifth byte (Event driven information):



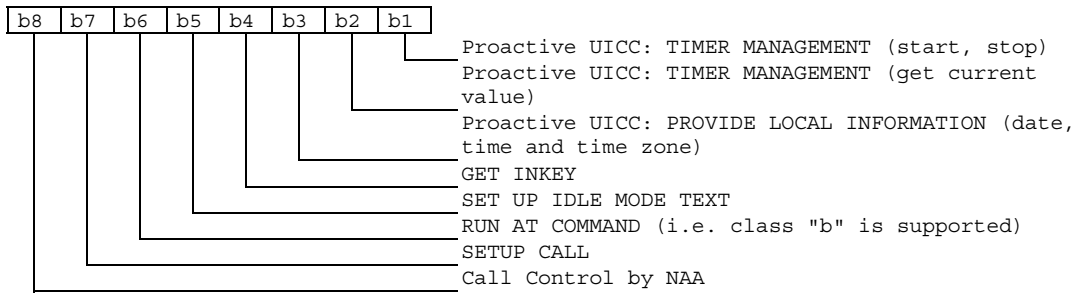
Sixth byte (Event driven information extensions):



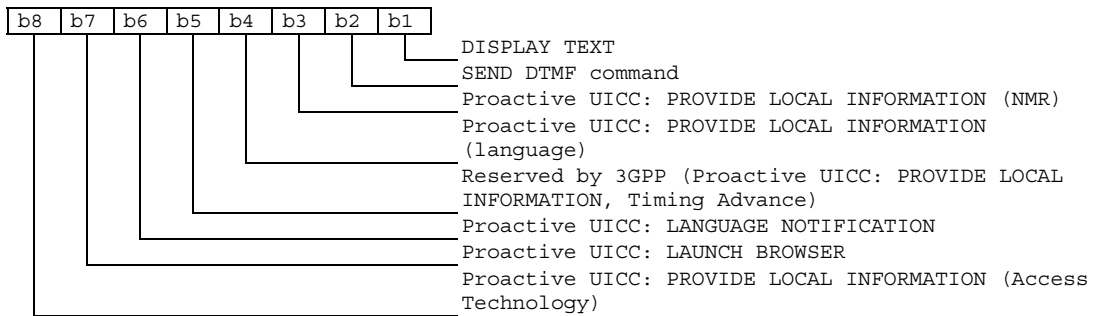
Seventh byte (Multiple card proactive commands) for class "a":



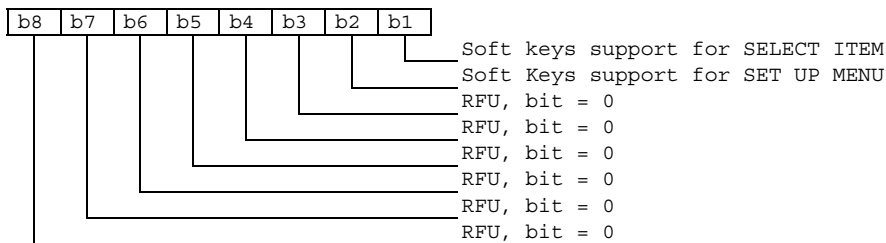
Eighth byte (Proactive UICC):



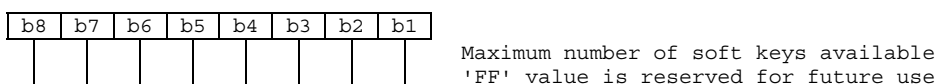
Ninth byte:



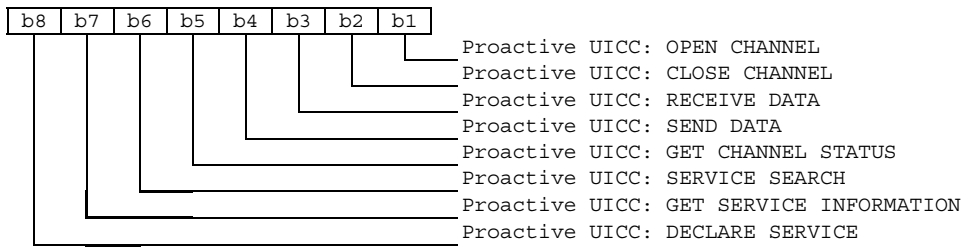
Tenth byte (Soft keys support) for class "d":



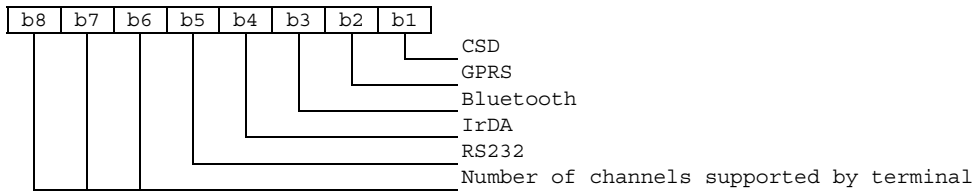
Eleventh byte (Soft keys information):



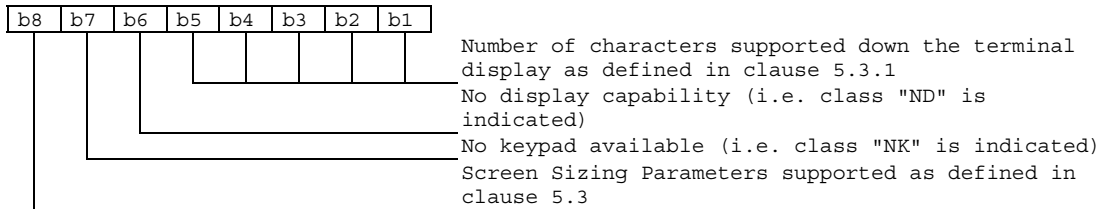
Twelfth byte (Bearer Independent protocol proactive commands, class "e"):



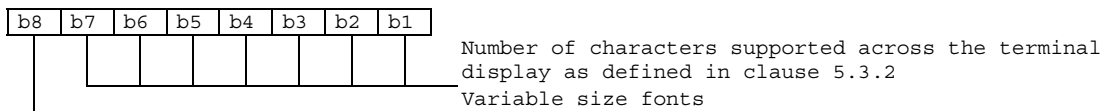
Thirteenth byte (Bearer Independent protocol supported bearers, class "e"):



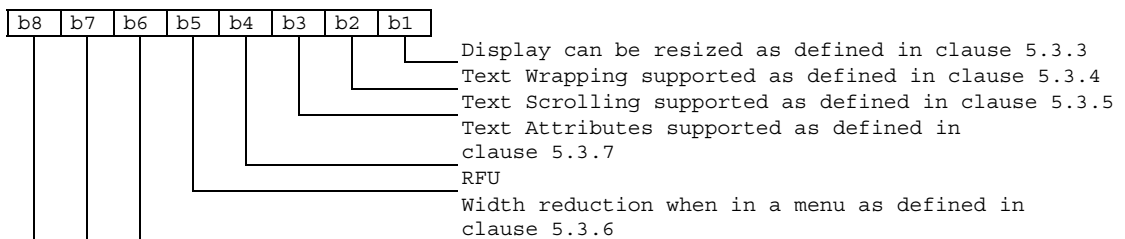
Fourteenth byte (Screen height):



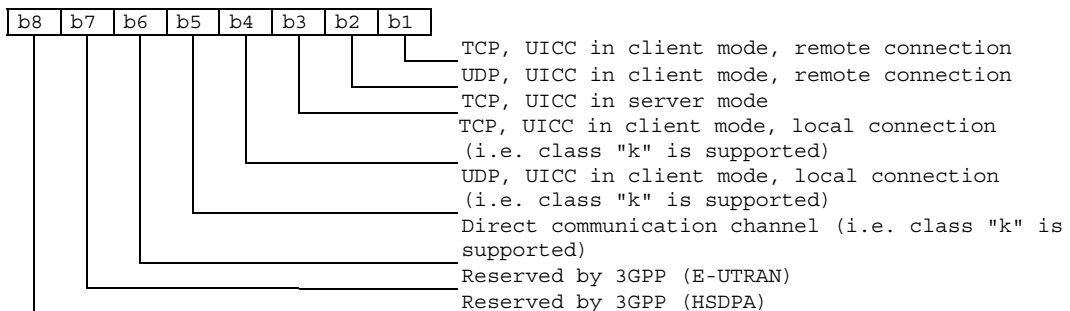
Fifteenth byte (Screen width):



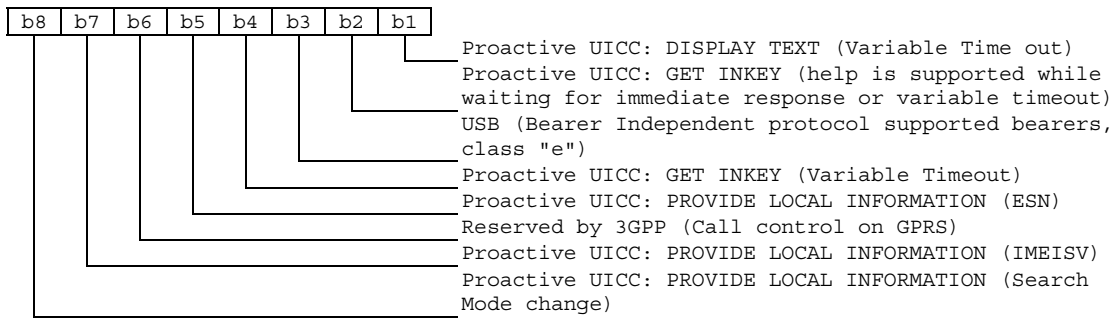
Sixteenth byte (Screen effects):



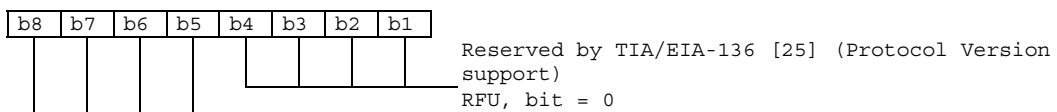
Seventeenth byte (Bearer independent protocol supported transport interface/bearers, class "e"):



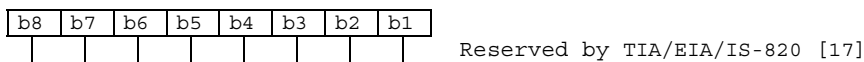
Eighteenth byte:



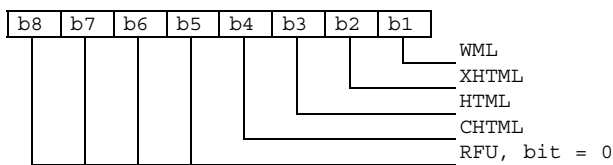
Nineteenth byte (reserved for TIA/EIA-136-C facilities [25]):



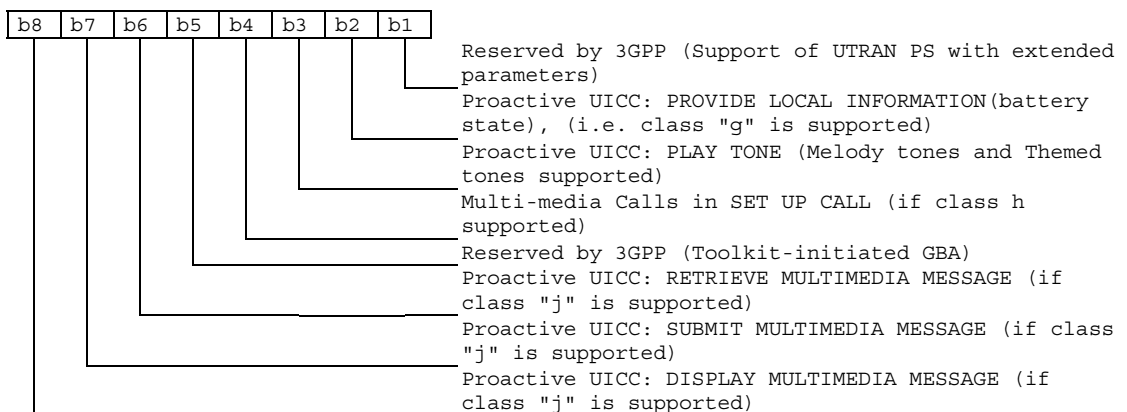
Twentieth byte (reserved for TIA/EIA/IS-820-A facilities [17]):



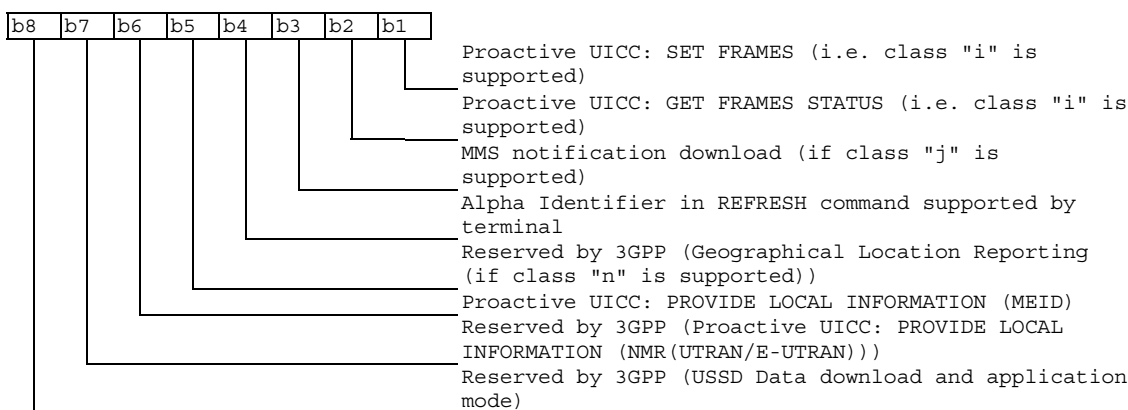
Twenty-first byte (Extended Launch Browser Capability) for class "c":



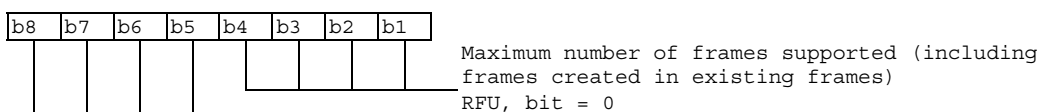
Twenty-second byte:



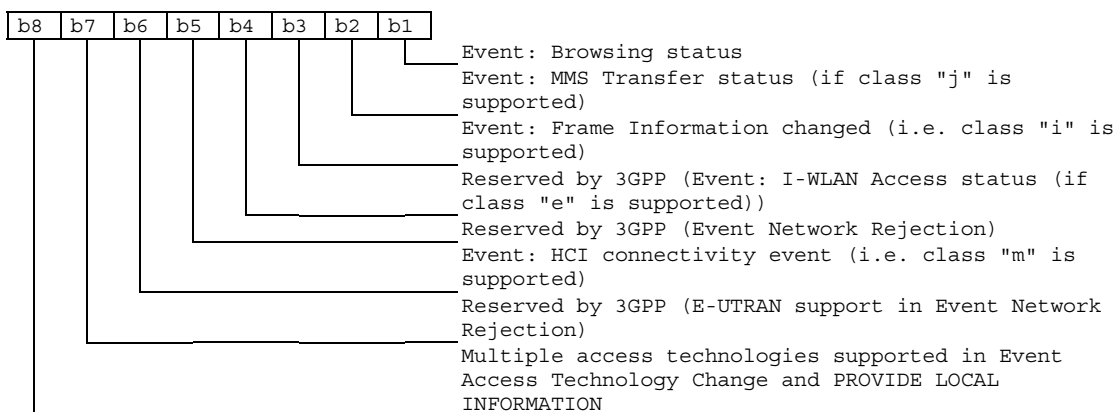
Twenty third byte:



Twenty fourth byte for class "i":

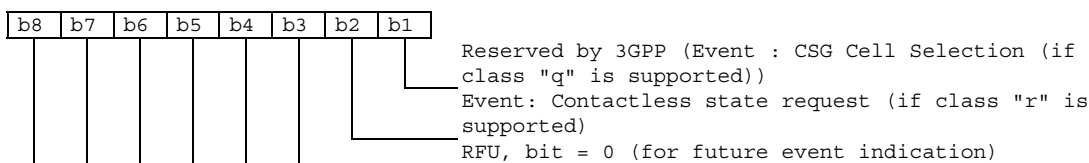


Twenty-fifth byte (Event driven information extensions):

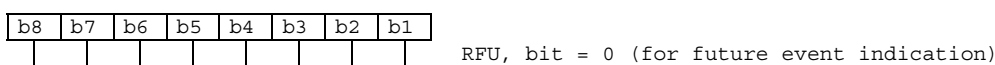


If bit "Multiple access technologies supported" is set to 1, it applies to the Event Access Technology Change if supported and all relevant modes of proactive command PROVIDE LOCAL INFORMATION that are supported.

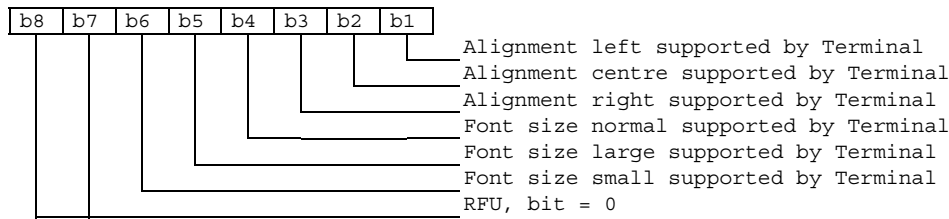
Twenty-sixth byte (Event driven information extensions):



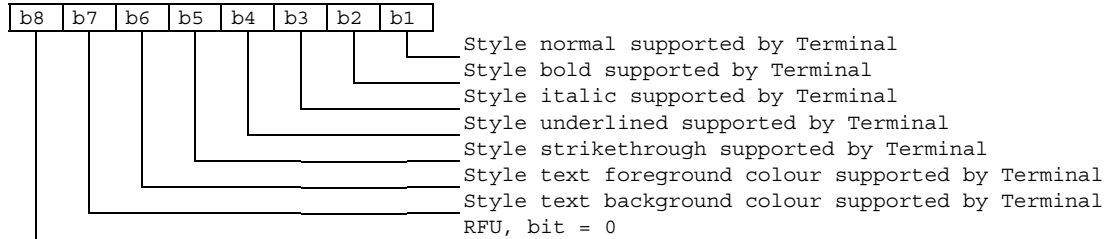
Twenty-seventh byte (Event driven information extensions):



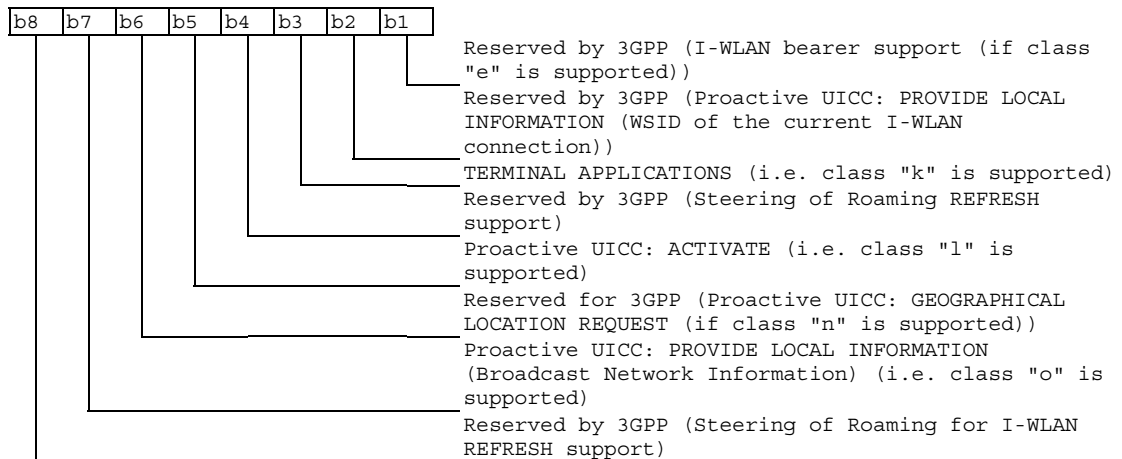
Twenty-eighth byte (Text attributes):



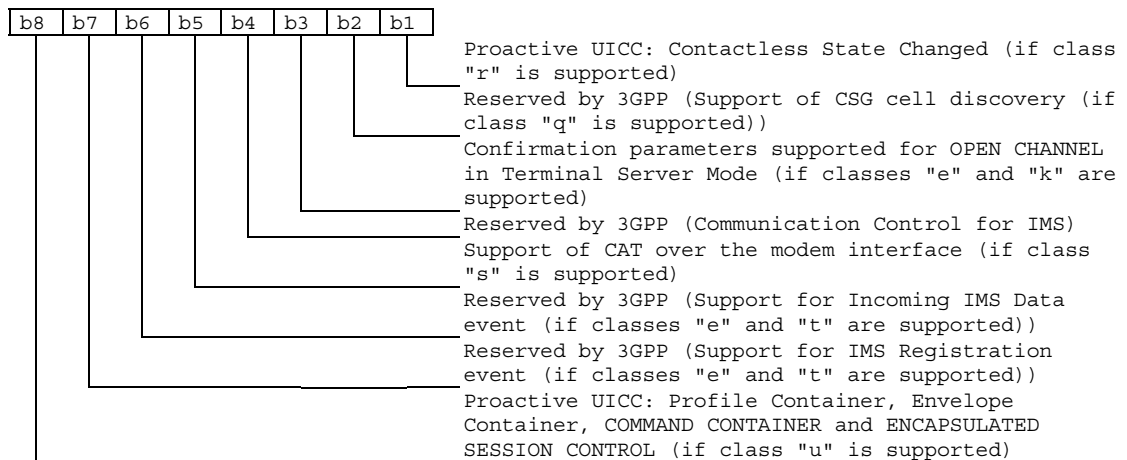
Twenty-ninth byte (Text attributes):



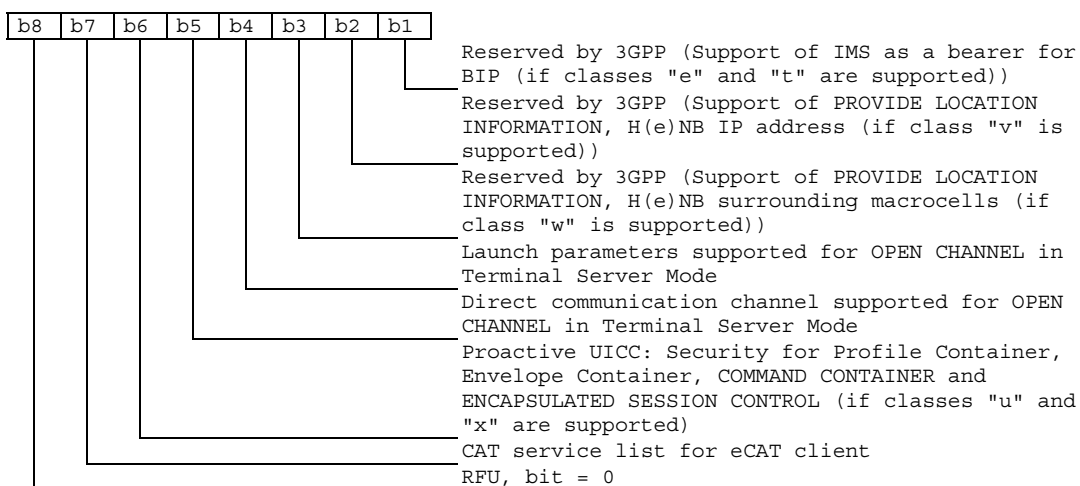
Thirtieth byte:



Thirty first byte:

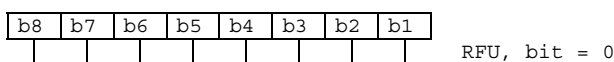


Thirty second byte:



The bit for CAT service list for eCAT client is only applicable for an eCAT client profile. It shall not be set in the command defined in this clause.

Subsequent bytes:



- RFU bits, and all bits of subsequent bytes, are reserved to indicate future facilities. A UICC supporting only the features of Card Application Toolkit defined here shall not check the value of RFU bits.
- Response parameters/data: None.

### 5.3 Definition of display parameters in profile download

This clause defines the terms used for defining the passing of the terminal's screen parameters from the terminal to the UICC.

#### 5.3.1 Number of characters supported down the terminal display

This is the guaranteed number of characters supported down the terminal display without scrolling (using the default character set specified in TS 123 038 [3]) as a result of a Display Text Proactive command.

If the screen resized as defined in clause 5.3.3 then this value shall be the initial number of characters supported before the display can be resized.

#### 5.3.2 Number of characters supported across the terminal display

This is the guaranteed number of characters supported across the terminal display without scrolling (using the default character set specified in TS 123 038 [3]) as a result of a Display Text Proactive command that can be viewed in one instance.

If the screen resized as defined in clause 5.3.3 then this value shall be the initial number of characters supported before the display can be resized.

### 5.3.3 Display can be resized

Display resize is supported if either:

- the user can change the number of characters supported across the display, down the display or both;
- the terminal can dynamically change the number of characters supported across the display, down the display or both.

### 5.3.4 Text wrapping

Text wrapping is supported if the terminal puts words that would be split across two lines, due to the display size, at the beginning of the next line down.

For class "i", if text wrapping is supported, it shall apply to the frames.

### 5.3.5 Text scrolling

Text scrolling is supported if the terminal scrolls, on one line, words that would be split across two lines, due to the display size.

For class "i", if text scrolling is supported, it shall apply to the frames.

### 5.3.6 Width reduction when in a menu

This value is the number of characters available across the display due to a DISPLAY TEXT proactive command without scrolling (using the default character set specified in TS 123 038 [3]) minus the number of characters available across the display due to a SELECT ITEM proactive command without scrolling (using the default character set specified in TS 123 038 [3]).

If the screen resized as defined in clause 5.3.3, then this value shall be calculated using the initial number of characters supported before the display can be resized.

### 5.3.7 Text attributes

Text that is displayed on the terminal screen can be displayed in various formats if the terminal supports it. If a Terminal receives a text attribute that it does not support then it shall use the default text attribute it supports.

**NOTE:** If the terminal supports text foreground colour and not text background colour or vice versa, and UICC requests terminal to display text of colour similar to the default text background colour of the terminal or vice versa, the text will not be distinguishable on the terminal screen.

A description of the various text formats are defined in TS 123 040 [27].

For class "i", if text attributes are supported, it shall apply to the frames.



---

## 6 Proactive UICC

### 6.1 Introduction

TS 102 221 [1], the 3G platform, defines that the terminal communicates to the UICC using the T=0 or T=1 protocols, which are specified in ISO/IEC 7816-3 [13]. Such communication may take place either over the electrical interface defined in TS 102 221 [1], or using the Smart Card functional interface over USB specified in TS 102 600 [38]. The terminal is always the "master" and initiates commands to the UICC, and therefore there is no mechanism for the UICC to initiate a communication with the terminal. This limits the possibility of introducing new UICC features requiring the support of the terminal, as the terminal needs to know in advance what actions it should take.

TS 151 011 [8], the 2G platform, defines that the terminal communicates to the SIM using the T=0 protocol, which is specified in ISO/IEC 7816-3 [13].

The UICC shall execute all CAT Proactive commands or procedures in such a way as not to jeopardize, or cause suspension, of service provisioning to the user. This could occur if, for example, execution of INTERNAL AUTHENTICATE is delayed by internal CAT activity, which would result in the network denying or suspending service to the user. Specifically, the MORE TIME command shall be used, whenever possible, to allow the terminal access to the 3G or 2G functionality of the UICC if a CAT application is taking an unreasonable amount of time to complete execution.

NOTE 1: The maximum work waiting time without sending a MORE TIME command depends on several factors (e.g. the permissible duration of a network-UICC authentication); in some cases as little as 2 s could be required. During this period the UICC should respect the work waiting time procedure, defined in TS 102 221 [1] and TS 151 011 [8].

NOTE 2: The use of frames does not allow several proactive commands to be executed in parallel.

The proactive UICC service provides a mechanism which stays within the application layer, but adds a new status response word SW1. This status response has the same meaning as the normal ending ('90 00'), and can be used with most of the commands that allow the normal ending, but it also allows the UICC to say to the terminal "I have some information to send to you". The terminal then uses the FETCH function to find out what this information is.

To avoid cross-phase compatibility problems, these functions shall only be used between a proactive UICC and a terminal that supports proactive UICC commands (see clause 6.2).

The UICC can issue a variety of commands through this mechanism, given in alphabetical order:

- **ACTIVATE:** which requests the terminal to activate a specified interface, e.g. the UICC-CLF interface (if class "l" is supported);
- **CLOSE CHANNEL:** which requests the terminal to close the specified data channel (if class "e" is supported);
- **COMMAND CONTAINER:** which sends a CAT command to an eCAT client;
- **CONTACTLESS STATE CHANGED:** which informs the terminal when the contactless functionality in the UICC has been enabled or disabled;
- **DECLARE SERVICE:** which requests the terminal to add or remove a service from its service database (the list of the resources available through a local bearer) (if class "f" is supported);
- **DISPLAY MULTIMEDIA MESSAGE:** which displays multimedia message on screen (if class "j" is supported);
- **DISPLAY TEXT:** which displays text or an icon on screen. A high priority is available, to replace anything else on screen;
- **ENCAPSULATED SESSION CONTROL:** which allows the UICC to end an encapsulated command session with an eCAT client and to setup and terminate an eCAT secure channel;

- **GET CHANNEL STATUS:** which requests the terminal to return the current status of all available data channels (if class "e" is supported);
- **GET FRAMES STATUS:** which requests the terminal to return the current parameters of all frames created (if class "i" is supported);
- **GET INKEY:** which sends text or an icon to the display and requests a single character response in return. It is intended to allow a dialogue between the UICC and the user, particularly for selecting an option from a menu;
- **GET INPUT:** which sends text or an icon to the display and requests a response in return. It is intended to allow a dialogue between the UICC and the user;
- **GET READER STATUS:** which gives information about the additional reader(s) and inserted card(s) (Card x state, e.g. powered on or not, Card x Presence), if class "a" is supported;
- **GET SERVICE INFORMATION:** which requests the terminal to look for detailed information on a given service on a given device (if class "f" is supported);
- **LANGUAGE NOTIFICATION:** which allows the UICC to notify the terminal about the currently used language in text strings issued by the CAT application;
- **LAUNCH BROWSER:** which requests a browser inside a browser enabled terminal to interpret the content corresponding to an URL;
- **MORE TIME:** which does not request any action from the terminal. The terminal is required to respond with TERMINAL RESPONSE (OK) as normal - see below. The purpose of the MORE TIME command is to provide a mechanism for the CAT task in the UICC to request more processing time;
- **OPEN CHANNEL:** which requests the terminal to open a data channel with parameters indicated in the command (if class "e" is supported);
- **PERFORM CARD APDU:** which requests the terminal to send an APDU command to the additional card, if class "a" is supported. This command is compatible with any protocol between the terminal and the additional card;
- **PLAY TONE:** which requests the terminal to play a tone in its earpiece, ringer, or other appropriate loudspeaker;
- **POLL INTERVAL:** which negotiates how often the terminal sends STATUS commands to the UICC during idle mode. Polling is disabled with **POLLING OFF**. Use of STATUS for the proactive UICC is described in TS 102 221 [1] for 3G platform and in TS 151 011 [8] for a 2G platform;
- **POWER OFF CARD:** which closes the session with the additional card, if class "a" is supported;
- **POWER ON CARD:** which initiates a session with the additional card and returns all the ATR bytes, if class "a" is supported;
- **PROVIDE LOCAL INFORMATION:** which requests the terminal to pass local information to the UICC, for example the mobile country and network codes (MCC + MNC) of the network on which the user is registered;
- **RECEIVE DATA:** which requests the terminal to return to the UICC data received on the specified channel (if class "e" is supported);
- **REFRESH:** which requests the terminal to carry out an initialization, and/or advises the terminal that the contents or structure of eFs on the UICC have been changed. The command also makes it possible to restart a card session by performing a reset;
- **RETRIEVE MULTIMEDIA MESSAGE:** which retrieves a Multimedia Message from the network (if class "j" is supported);
- **RUN AT COMMAND:** which will convey an AT Command to the terminal, and cause the response to the AT Command to be returned to the UICC;

- **SELECT ITEM:** where the UICC supplies a list of items, and the user is expected to choose one. The terminal presents the list in an implementation-dependent way;
- **SEND DATA:** which requests the terminal to send on the specified channel data provided by the UICC (if class "e" is supported);
- **SEND DTMF:** which requests the terminal to send DTMF tone(s) during an established call;
- **SEND SHORT MESSAGE:** which sends a short message or SMS-COMMAND to the network;
- **SERVICE SEARCH:** which requests the terminal to look for services available in the terminal environment (if class "f" is supported);
- **SET FRAMES:** which requests the terminal to set frames on the screen with parameters indicated in the command (if class "i" is supported);
- **SET UP CALL:** of which there are three types:
  - set up a call, but only if not currently busy on another call;
  - set up a call, putting all other calls (if any) on hold;
  - set up a call, disconnecting all other calls (if any).
- **SET UP EVENT LIST:** where the UICC supplies a list of events which it wants the terminal to provide details of when these events happen;
- **SET UP IDLE MODE TEXT:** which supplies a text string to be used by the terminal as stand-by mode text;
- **SET UP MENU:** where the UICC supplies a list of items to be incorporated into the terminal's menu structure;
- **SUBMIT MULTIMEDIA MESSAGE:** which sends a Multimedia Message to the network (if class "j" is supported).
- **TIMER MANAGEMENT:** which requests the terminal to manage a timer in a way described in the command (start, deactivate and get the current value) and, in the case of starting a timer, for a duration indicated in the command.

The terminal tells the UICC if the command was successful or not using the command result procedure defined in clause 6.7. Responsibility for what happens after that (whether to repeat the command, try another one immediately, try again sometime later, or not to try again at all) lies with the CAT. However, the CAT needs to know why the command failed, so the terminal provides the UICC with the result of the command.

Results are grouped into three main types:

- OK;
- temporary problem. These results are further broken down into types of temporary problems, and specific causes. Generally, they indicate to the UICC that it may be worth trying again;
- permanent problem. These results are again further broken down into types of permanent problems, and specific causes. Generally, they indicate to the UICC that it is not worth trying again during this card session.

If the UICC issues an instruction to the terminal to initiate a terminal Originated transaction (e.g. SEND SMS, SEND DTMF), then unless explicitly stated elsewhere in the present document or in TS 102 221 [1] or TS 151 011 [8], the content supplied by the UICC for onward transmission by the terminal shall not be altered by the terminal.

## 6.2 Identification of terminal support

A terminal that supports proactive UICCs shall be identified as such when it sends a **TERMINAL PROFILE** command during UICC initialization. A proactive UICC shall not send any command requests (status bytes SW1 SW2 = '91 XX') to a terminal that does not support the proactive UICC feature.

## 6.3 General procedure

For all of the procedures that can end in '90 00' (indicating normal ending to the command) a proactive UICC operating with a terminal that supports proactive UICCs may instead use the status response '91 XX'.

The response code '91 XX' shall indicate to the terminal that the previous command has been successfully executed by the UICC in the same way as '90 00' (i.e. "OK"), but additionally it shall indicate response data which contains a command from the UICC for a particular terminal procedure (defined in clause 6.4).

The value 'XX' indicates the length of the response data. The terminal shall use the FETCH command to obtain this data.

It is the responsibility of the UICC to remind the terminal of a pending proactive command by applying the '91 XX' return code until it is fetched by the terminal.

NOTE 1: The last value of 'XX' received in a '91 XX' return code from the UICC should be used by the terminal in a following FETCH command.

NOTE 2: It is recommended that the terminal interprets a '90 00' following a '91 XX' without a corresponding FETCH as if no proactive command is available in the UICC and regard the proactive UICC session as being terminated. However, the UICC should be able to handle a FETCH command being sent in this case, e.g. by applying the appropriate error handling (cf. "Handling of unknown, unforeseen and erroneous messages").

TS 102 221 [1] and TS 151 011 [8] show how the UICC and the SIM can initiate a proactive command.

When the terminal has received a command from the UICC, it shall attempt to process the command immediately:

- if the command has been successfully executed, the terminal shall inform the UICC as soon as possible, using TERMINAL RESPONSE except when specified otherwise in the present document (e.g. see clause 6.4.7);
- if the command was not successfully executed, the terminal shall inform the UICC as soon as possible using TERMINAL RESPONSE with an error condition.

Responsibility for re-trying lies with the UICC application. The CAT can make a judgement whether to send the same command again, to send a different one, or not to try again, from the information given by the terminal in TERMINAL RESPONSE. If the UICC application wishes the terminal to try again, it shall issue a new (identical) command.

Only one proactive command can be ongoing at any one time.

## 6.4 Proactive UICC commands and procedures

### 6.4.1 DISPLAY TEXT

For a terminal of type ND the support of this command is optional.

This command instructs the terminal to display a text message, and/or an icon (see clause 6.5.4). It allows the UICC to define the priority of that message, and the text string format.

Two types of priority are defined:

- display normal priority text and/or icon on screen;
- display high priority text and/or icon on screen.

The text string can be in one of three formats:

- packed format in SMS default alphabet (see clause 8.15.2);
- unpacked format in SMS default alphabet (see clause 8.15.2);
- UCS2 alphabet format (see clause 8.15.3).

NOTE 1: The text string may contain up to 240 bytes.

A flag (see command qualifier, clause 8.6) shall be set to inform the terminal whether the availability of the screen for subsequent information display after its use for "Display Text" should be either after a short delay (the duration of the delay being at the discretion of the terminal manufacturer unless an exact duration is indicated by a duration object), or following a user MMI action.

An immediate response object may be included by the UICC, to indicate if the terminal should sustain the display beyond sending the TERMINAL RESPONSE. Terminal support of this feature is mandatory, if DISPLAY TEXT is supported.

A duration object that represents the variable display timeout may be included by the UICC. The duration informs the terminal about the required duration of the display (Precision and resolution are in accordance with clause 6.4.21 Timer Management). The requested timeout value replaces the timeout set by the terminal manufacturer. Terminal support of this feature is indicated in the PROFILE DOWNLOAD. The behaviour of terminals that do not support this feature is dependent on the Comprehension Required flag:

- if the user has indicated the need to end the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with "Proactive UICC session terminated by the user" result value;
- if the user has indicated the need to go backwards in the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with "Backward move in the proactive UICC session requested by the user" result value;
- if a flag of the command qualifier (see clause 8.6) indicates that the terminal shall wait for the user to clear message and if the terminal decides that no user response has been received, the terminal shall send a TERMINAL RESPONSE with "No response from user" result value. A terminal of type NK which receives a DISPLAY TEXT command with a command qualifier flag indicating "DISPLAY TEXT - wait for user to clear" shall send a TERMINAL RESPONSE with "Command performed successfully" result value;
- if the UICC includes a duration object, the terminal shall limit the display time of the message for a period that does not exceed the requested duration. The timer starts when the text is displayed on the screen and stops when the TERMINAL RESPONSE is sent except if the text is to be sustained beyond an immediate response. The timeout may be used with other options of this command. The variable timeout does not affect TERMINAL RESPONSE values that are deriving from other chosen options of this command;
- if the UICC includes an immediate response object, the terminal shall immediately send TERMINAL RESPONSE (Command performed successfully). The terminal shall continue to display the text until one of the following events occurs:
  - a subsequent proactive command is received containing display data;
  - the expiration of the variable display timeout, if so indicated by the duration object;
  - the expiration of the short delay, if so indicated by the command qualifier;
  - following a user MMI action;
  - when a higher priority event occurs, e.g. an incoming mobile terminated call.
- no further TERMINAL RESPONSE shall be sent when the terminal removes the text from the display, regardless of the cause;
- otherwise, the terminal shall send TERMINAL RESPONSE (Command performed successfully) at the expiration of either the short delay or the variable display timeout, or following a user MMI action not described above.

In each case the availability of the screen for the subsequent information display is defined in clause 6.9.

NOTE 2: For the case where the text is cleared after a short delay, the terminal may also allow the user to clear the display via the MMI prior to this.

The terminal shall reject normal priority text commands if the screen is currently being used for more than its normal stand-by display. If the command is rejected, the terminal informs the UICC using TERMINAL RESPONSE (terminal currently unable to process command - screen busy).

High priority text shall be displayed on the screen immediately, except if there is a conflict of priority level of alerting such as incoming calls or a low battery warning. In that situation, the resolution is left to the terminal. If the command is rejected in spite of the high priority, the terminal shall inform the UICC using TERMINAL RESPONSE (terminal currently unable to process command - screen is busy).

If help information is requested by the user, this command may be used to display help information on the screen. The help information should be sent as high priority text and with the option that it should be cleared after a short delay.

## 6.4.2 GET INKEY

For a terminal of type ND or NK the support of this command is optional.

This command instructs the terminal to display text and/or an icon (see clause 6.5.4) and to expect the user to enter a single character. Any response entered by the user shall be passed transparently by the terminal to the UICC.

The text can be in one of three formats:

- packed format in SMS default alphabet (see clause 8.15.2);
- unpacked format in SMS default alphabet (see clause 8.15.2);
- UCS2 alphabet format (see clause 8.15.3).

The response can be from one of three character sets. This is specified by the UICC:

- digits only (0 to 9, \*, #, and +);
- characters from the SMS default alphabet;
- characters from the UCS2 alphabet.

Upon receiving the command, the terminal shall display the text. The terminal shall allow the user to enter a single character in response:

- if the user has indicated the need to go backwards in the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with "Backward move in the proactive UICC session requested by the user" result value;
- if the user has indicated the need to end the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with "Proactive UICC session terminated by the user" result value;
- if the terminal decides that no user response has been received, the terminal shall send a TERMINAL RESPONSE with "No response from user" result value;
- if the UICC requests an immediate digit response, the terminal shall only allow the user to enter a character that can be entered by a single key press (that means for terminals providing only the keypad as defined in TS 122 030 [28], from the digits 0 to 9, \* and # (but not +)). When the user has entered a digit, the terminal shall pass the entered digit transparently to the UICC, using TERMINAL RESPONSE. The terminal shall not display the entered digit in any way. The terminal shall not allow the user to change the entered digit. The terminal shall not request the user to confirm the response.

NOTE 1: A larger portion of the screen may be used for display purposes, since the terminal shall not display the entered digit in any way.

- if the UICC requests a digit only, the terminal shall only allow the user to enter a character from the digits 0 to 9, \*, # and +. When the user has entered a digit, the terminal shall pass the entered digit transparently to the UICC, using TERMINAL RESPONSE;

- if help information is available for the command and if the user has indicated the need to get help information, the terminal shall send a `TERMINAL RESPONSE` with "help information required by the user" result value. Depending of terminal implementation, combination with the option "immediate response" and/or the option "variable timeout" may result that the user is unable to request the help;
- the terminal support of help information combined with immediate response and/or timeout is indicated in the `TERMINAL PROFILE`;
- if the UICC requests a character from the SMS default alphabet, the terminal shall allow the user to enter a character using characters from this alphabet. When the user has entered a character, the terminal shall pass the entered character transparently to the UICC, using `TERMINAL RESPONSE`;
- if the UICC requests a "Yes/No" response, the terminal shall allow the user to enter either a positive or a negative decision using MMI means left to terminal manufacturer's choice (keypad, touch screen, softkey, etc.). The terminal may use `SEND`, `ACCEPT` or `END` functions in relation to `GET INKEY "Yes/No"` response. If used, the `SEND` and `ACCEPT` functions as defined in TS 122 030 [28] shall mean positive decision and the `END` function as defined in TS 122 030 [28] shall mean a negative one. Depending on the user's choice, the terminal shall pass the positive or a negative value to the UICC, using `TERMINAL RESPONSE`;
- if the UICC requests a "Yes/No" response together with immediate digit response, the terminal shall combine the behaviour of "Yes/No" UICC request with the behaviour of an immediate digit response UICC request;
- if the UICC requests a variable timeout, the terminal shall wait until either the user enters a single character or the timeout expires. The timer starts when the text is displayed on the screen and stops when the `TERMINAL RESPONSE` is sent. The terminal shall pass the total display text duration (command execution duration) to the UICC using the `TERMINAL RESPONSE`. The time unit of the response is identical to the time unit of the requested variable timeout. The timeout may be used with other options of this command. The variable timeout does not affect `TERMINAL RESPONSE` values that are deriving from other chosen options of this command. terminal support of this feature is indicated in the `PROFILE DOWNLOAD`. The behaviour of terminals that do not support this feature is dependent on the `Comprehension Required` flag.

NOTE 2: If the MMI of the terminal requires more than one key press in order to select a character, it is an implementation decision for the terminal manufacturer how to indicate completion (e.g. timeout, pressing `SEND`, `OK`). It may be useful to echo the input character on the display.

For digits only (0 to 9, \*, # and +) and SMS default alphabet characters sets, the response shall be coded using the SMS default alphabet in unpacked format.

### 6.4.3 GET INPUT

For a terminal of type ND or NK the support of this command is optional.

This command instructs the terminal to display text and/or an icon (see clause 6.5.4) and that any response string entered by the user shall be passed transparently by the terminal to the UICC. If the UICC provides a default text, the terminal shall display this default text, which the user may accept, reject or edit as the response string.

The text can be in one of three formats:

- packed format in SMS default alphabet (see clause 8.15.2);
- unpacked format in SMS default alphabet (see clause 8.15.2);
- UCS2 alphabet format (see clause 8.15.3).

The UICC indicates how many characters are expected for the response string, by giving a minimum and a maximum acceptable length.

The UICC specifies the following variables for the response string it is expecting from the user:

- the response contains either digits only (0 to 9, \*, # and +) or characters from one of the possible alphabets;
- the response contains either characters coded in SMS default alphabet or characters coded in UCS2 alphabet;

- the response for digits only (0 to 9, \*, # and +) or characters from SMS default alphabet is either in an unpacked format or in a packed format;
- the terminal may display the text string being entered by the user (the response), or the terminal shall hide the actual text string.

The combination of characters from either the SMS default alphabet or the UCS2 alphabet and hidden entry mode is not allowed. In hidden entry mode, only digits from the set "0 to 9", "\*" and "#" are allowed for the user input. "+" is not allowed for user input in this mode.

If the UICC requests that the user input (text string) is to be hidden, the terminal shall prevent the text string from being identified by any means. For example, the text string shall not be displayed and no DTMF tones shall be emitted. Nevertheless, it is permissible for the terminal to indicate the entry of characters, so long as the characters themselves are not revealed.

If the terminal supports the predictive text feature it shall be used in the same context as normal text entry/editing within the response string entered by the user unless the hidden entry mode has been requested by UICC.

Upon receiving the command, the terminal shall display the text. The terminal shall allow the user to enter characters in response:

- the terminal MMI is responsible for managing the entry of the correct number of characters;
- if the user has indicated the need to go backwards in the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with "Backward move in the proactive UICC session requested by the user" result value;
- if the user has indicated the need to end the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with "Proactive UICC session terminated by the user" result value;
- if the terminal decides that no user response has been received, the terminal shall send a TERMINAL RESPONSE with "No response from user" result value;
- if the UICC requests digits only, the terminal shall only allow the user to enter the digits 0 to 9, \*, # and +. When the user has indicated completion, the terminal shall pass the entered digit string transparently to the UICC, using TERMINAL RESPONSE;
- if the UICC requests characters from the UCS2 alphabet or SMS default alphabet, the terminal shall allow the user to enter a character string using characters from one of these alphabets. When the user has indicated completion, the terminal shall pass the entered text string transparently to the UICC, using TERMINAL RESPONSE;
- if help information is available for the command and if the user has indicated the need to get help information, the terminal shall send a TERMINAL RESPONSE with "help information required by the user" result value;
- if the UICC requests the user input to be in packed format, then the terminal shall pack the text according to TS 123 038 [3] before submitting it to the UICC.

#### 6.4.4 MORE TIME

This procedure is provided to allow the CAT task in the UICC more time for processing, where the processing is so long that it is in danger of affecting normal operation, and clock stop prevents processing to take place in the background.

The terminal shall take no extraordinary action when it receives this command, and all other operations shall be unaffected. The terminal shall conclude the command by sending TERMINAL RESPONSE (OK) to the UICC, as soon as possible after receiving the MORE TIME command.



## 6.4.5 PLAY TONE

For a terminal of type NA the support of this command is optional.

This command instructs the terminal to play an audio tone.

Upon receiving this command, the terminal shall check if it is currently in, or in the process of setting up (SET-UP message sent to the network, see TS 124 008 [20]), a speech call:

- if the terminal is in, or is setting up a speech call, it shall superimpose the tone on top of the downlink audio (if any), for the duration given in the command. The progress or current state of the call shall not be affected in any way. The terminal shall send the TERMINAL RESPONSE (Command performed successfully) as soon as possible after the tone has been completed and, if an alpha identifier was included and displayed, the screen is available for subsequent information display;
- if the terminal is not in or setting up a speech call, it shall route the audio to the external ringer, or other appropriate audio device, and play the tone for the duration given in the command. The terminal shall send the TERMINAL RESPONSE (Command performed successfully) as soon as possible after the tone has been completed and, if an alpha identifier was included and displayed, the screen is available for subsequent information display.

The terminal shall additionally follow the general behaviour upon receiving this command:

- if the user has indicated the need to end the proactive UICC session while the terminal plays the tone, the terminal shall stop playing the tone and shall send a TERMINAL RESPONSE with "Proactive UICC session terminated by the user" result value;
- if terminal support for the specific tone requested is optional, and the terminal does not support this particular tone, the terminal shall inform the UICC using TERMINAL RESPONSE (Command beyond terminal's capabilities);
- if the terminal supports a tone, but it is not possible to play it, the terminal shall inform the UICC using the TERMINAL RESPONSE (Command performed successfully, tone not played).

Terminal shall not generate any verbal indication or display any text or graphical indication about the normal meaning of this tone (e.g. display "called subscriber busy"). If the UICC wishes to convey a meaning in text to the user, it shall do this through the alpha identifier data object and/or an icon (see clause 6.5.4).

The use of this alpha identifier by the terminal is described below:

- if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it to inform the user. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
- if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), the terminal should not give any information to the user;
- if the alpha identifier is not provided by the UICC, the terminal may give information to the user concerning what is happening.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

If the terminal is required to generate a supervisory tone due to the progress of the current call (e.g. the network sends the terminal call control cause information) as defined in TS 122 001 [2], then the call supervisory tone shall take precedence over the tone requested by the UICC.

## 6.4.6 POLL INTERVAL

This procedure negotiates how often the terminal shall send STATUS commands related to Proactive Polling (defined in TS 102 221 [1] and in TS 151 011 [8]). The UICC indicates the poll interval it requests from then onwards, and the terminal responds through TERMINAL RESPONSE with the maximum interval that it will use. If the terminal does not support the poll interval requested by the UICC, then the terminal shall respond with the closest interval to the one requested by the UICC, or, if the intervals the terminal can offer are equidistant (higher and lower) from the UICC's request, the terminal shall respond with the lower interval of the two.

Applications on the UICC should not request short time intervals for an extended period, as this will have an adverse effect on battery life, and should not use this command for time management purposes.

## 6.4.7 REFRESH

The purpose of this command is to enable the terminal to be notified of the changes to the UICC configuration that have occurred as the result of a NAA application activity. It is up to the NAA application to ensure that this is done correctly.

On a 3G platform, the UICC may indicate the AID of the NAA application it wants to REFRESH:

- if the indicated NAA is active, the terminal shall perform the REFRESH;
- if indicated NAA is not active, the terminal shall send TERMINAL RESPONSE: REFRESH performed but indicated NAA was not active. The terminal shall not select the indicated NAA;

NOTE 0: If the indicated NAA is not active, the terminal takes no action on the refresh command except sending the TERMINAL RESPONSE as indicated.

- if no AID is indicated, then the terminal shall assume the REFRESH applies to the NAA application currently selected on the basic logical channel (logical channel 0). If no NAA is currently selected on the basic logical channel, the terminal shall send a TERMINAL RESPONSE (command performed successfully).

A 2G platform shall not provide the AID COMPREHENSION-TLV.

The command supports seven different modes:

- NAA Initialization. This mode tells the terminal to carry out NAA initialization as it is defined by the NAA, starting after the PIN verification procedure;
- NAA File Change Notification. This mode advises the terminal of the identity of the eFs that have been changed (in structure and/or contents) in the indicated NAA and files under DF<sub>TELECOM</sub>. This information can be used by the terminal if there is an image of NAA eFs in the terminal's memory, to determine whether it needs to update this image;
- NAA Initialization and File Change Notification. This is a combination of the first two modes above;
- NAA Initialization and Full File Change Notification. This mode causes the terminal to perform the NAA initialization procedure of the first mode above and advises the terminal that several eFs have been changed (in structure or contents) in the indicated NAA. If there is an image of NAA eFs in the terminal's memory, the terminal shall completely update this image;
- UICC Reset. This mode causes the terminal to run the application session termination procedure in accordance with TS 102 221 [1] for every active application. Subsequently, the terminal performs a reset (warm reset preferred) on the UICC and starts a new card session. The terminal shall not send the TERMINAL RESPONSE; this is an exception from the normal procedure, where TERMINAL RESPONSE is sent after completion of the command. The UICC shall interpret the reset as an implicit TERMINAL RESPONSE. The UICC Reset mode is used when a CAT requires ATR or complete UICC initialization procedures to be performed;
- NAA Application Reset. This mode causes the terminal to run the NAA session termination and the NAA application closure procedures in accordance with NAA specification. Subsequently, the terminal performs NAA initialization procedure. This mode is only applicable on a 3G platform, and shall not be used on a 2G platform;

- NAA Session Reset. This mode is equivalent to "NAA Initialization and File Change Notification" mode and in addition requires the terminal to perform a specific NAA procedure. This mode is only applicable on a 3G platform, and shall not be used on a 2G platform.

If the terminal performs the REFRESH command successfully for only those eFs indicated in the mode, the terminal shall inform the UICC using TERMINAL RESPONSE (OK), after it has completed its refreshing (i.e. taking into account the new value of the eFs).

For REFRESH commands with mode other than "UICC Reset" or "NAA Application Reset", it is permissible for the terminal, as part of its execution of the REFRESH command, to read eFs in addition to those notified by the UICC, or to perform a NAA initialization, provided that the procedure executed wholly encompasses the mode requested by the UICC and does not involve re-entering the PIN. The terminal shall not perform a reset. If the terminal does the refreshing successfully, it shall inform the UICC using TERMINAL RESPONSE (Refresh performed with additional eFs read), after the terminal has completed its refreshing. It should be noted that reading additional eFs will lengthen the refresh procedure.

For REFRESH command with mode "UICC Reset", the UICC can not start a proactive session before profile download is executed after the reset.

NOTE 1: For REFRESH command with mode "NAA Application Reset", it is not necessary for the terminal to send the TERMINAL PROFILE after the application reset.

If the terminal receives a REFRESH command while in a state where execution of the command would be unacceptable, upsetting the current user operation (e.g. notification during a call that the IMSI has changed), the terminal shall inform the UICC using TERMINAL RESPONSE (terminal currently unable to process command - currently busy on call) or TERMINAL RESPONSE (terminal currently unable to process command - screen is busy) as appropriate.

NOTE 2: Many terminals copy an image of the NAA application files to the terminal at initialization to speed up access to these fields during a NAA session. One of the purposes of this coding of the REFRESH command is to enable terminals to change such an image efficiently.

If, on receipt of the REFRESH command, the terminal replies that it is busy (e.g. in call or navigating menus), the toolkit application may retry it later.

It is recommended for the terminal to minimize the use of sending temporary problem TERMINAL RESPONSE, as during the period between the UICC issuing a REFRESH command and the terminal performing the refresh procedure, there may be inconsistencies between data held in the terminal and in the UICC. However, responsibility for retrying of all pro-active commands lies with the UICC.

Optionally, the UICC may include in this command an alpha identifier. The use of this alpha identifier by the terminal is described below:

- if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it to inform the user. This is also an indication that the terminal should not give any other information to the user on the fact that the terminal is performing the refresh command. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
- if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not give any information to the user on the fact that the terminal is performing the refresh command;
- if the alpha identifier is not provided by the UICC, the terminal may give information to the user concerning what is happening.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

## 6.4.8 SET UP MENU

For a terminal of type ND or NK the support of this command is optional.

The UICC shall supply a set of menu items, which shall be integrated with the menu system (or other MMI facility) in order to give the user the opportunity to choose one of these menu items at his own discretion. Each item comprises a short identifier (used to indicate the selection), a text string and optionally an icon identifier, contained in an item icon identifier list data object located at the end of the list of items.

The UICC shall include an alpha identifier, and optionally an icon identifier, which acts as a title for the list of menu items. This icon may be used by the terminal to provide an entry into the list of toolkit menu items for the user.

If an icon is provided by the UICC, the icon(s) indicated in the command may be used by the terminal in addition to, or instead of the alpha identifier or text string, as indicated with the icon qualifier (see clause 6.5.4). Additionally if soft key preferred is indicated in the command details and soft key for SET UP MENU is supported by the terminal and the number of icons items does not exceed the number of soft keys available then the terminal shall display those icons as soft key.

The UICC may include an items next action indicator data object located at the end of the list of items. The inclusion of the items next action indicator is to allow the terminal to indicate to the user the consequences of performing the selection of an item.

**NOTE:** The maximum amount of data sent in one proactive UICC command is 256 bytes. It is therefore unavoidable that there is trade-off between the number of items and the length of the descriptive text (the alpha identifier of the SET-UP MENU command and the text strings of the items), e.g. for an average length of 10 bytes per text string the maximum amount of items is 18.

The list of menu items shall then be part of the menu system of the terminal and the user is allowed to select an item from this list. The presentation style is left as an implementation decision to the terminal manufacturer. However, the terminal shall present the menu items in the order given by the UICC, unless instructed otherwise by the user, or when this would be inappropriate for the presentation style of the terminal. The menu provided by the UICC in the last SET UP MENU command shall no longer be part of the menu system of the terminal if the terminal is powered off or the UICC is removed or a reset is performed.

Any subsequent SET-UP MENU command replaces the current list of menu items supplied in the previous SET-UP MENU command. The SET-UP MENU command can also be used to remove a menu from the menu system in the terminal, see clause 6.6.7.

When the terminal has successfully integrated or removed the list of menu items, it shall send TERMINAL RESPONSE (OK) to the UICC.

When the terminal is not able to successfully integrate or remove the list of menu items, it shall send TERMINAL RESPONSE (Command beyond terminal's capabilities).

When the user has selected one of the menu items of this menu item list, then the terminal shall use the Menu Selection mechanism to transfer the identifier of the selected menu item to the UICC.

If help is available for the command and if the user has indicated the need to get help information on one of the menu items, the terminal shall use the Menu Selection mechanism to inform the UICC about this help request.

## 6.4.9 SELECT ITEM

For a terminal of type ND or NK the support of this command is optional.

The UICC shall supply a set of items from which the user may choose one. Each item comprises a short identifier (used to indicate the selection), a text string and optionally an icon identifier, contained in an item icon identifier list data object located at the end of the list of items.

Optionally the UICC may include an alpha identifier, and an icon identifier. These are intended to act as a title for the list of items. The UICC may include an items next action indicator data object located at the end of the list of items. The inclusion of the items next action indicator is to allow the terminal to indicate to the user the consequences of performing the selection of an item.

The alpha identifier included by the UICC shall be used by the terminal as the title for the list of items.

If an icon is provided by the UICC, the icon(s) indicated in the command may be used by the terminal in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4). Additionally, if "selection using soft key preferred" is indicated in the command details and "soft key for SELECT ITEM" is supported by the terminal and the number of icons items does not exceed the number of soft keys available, then the terminal shall display those icons as soft keys.

**NOTE:** The maximum amount of data sent in one proactive UICC command is 256 bytes. It is therefore unavoidable that there is trade-off between the number of items and the length of the descriptive text (the alpha identifier of the SELECT ITEM command and the text strings of the items), e.g. for an average length of 10 bytes per text string the maximum amount of items is 18.

The terminal shall present the list of text strings to the user, and allow the user to select an item from this list. A flag of the command qualifier (see clause 8.6) indicates whether the list is a choice of navigation options, or a choice of data values. The presentation style is left as an implementation decision to the terminal manufacturer. However, the terminal shall present the menu items in the order given by the UICC, unless instructed otherwise by the user, or when this would be inappropriate for the presentation style of the terminal. The menu provided by the UICC in the last SET-UP MENU command shall no longer be part of the menu system of the terminal if the terminal is powered off or the UICC is removed or a reset is performed.

The UICC may supply with the list, if applicable, indication of the default item, e.g. the previously selected item.

When the user has selected an item, the terminal shall send TERMINAL RESPONSE (OK) to the UICC with the identifier of the item chosen:

- if the user has indicated the need to end the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with "Proactive UICC session terminated by the user" result value;
- if the user has indicated the need to go backwards in the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with "Backward move in the proactive UICC session requested by the user" result value;
- if the terminal decides that no user response has been received, the terminal shall send a TERMINAL RESPONSE with "No response from user" result value;
- if help information is available for the command and if the user has indicated the need to get help information, the terminal shall send a TERMINAL RESPONSE with "help information required by the user" result value to the UICC with the identifier of the item for which the user is requiring help information.

## 6.4.10 SEND SHORT MESSAGE

This command requests the terminal to send a short message. The type and coding of TPDU depends on the network technology.

Two types are defined:

- a short message to be sent to the network, where the user data can be passed transparently;
- a short message to be sent to the network where the text needs to be packed by the terminal.

Coding of the message and use of packing are defined by the appropriate and network technologies dependent specifications.

Optionally, the UICC may include in this command an alpha identifier. The use of this alpha identifier by the terminal is described below:

- if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it to inform the user. This is also an indication that the terminal should not give any other information to the user on the fact that the terminal is sending a short message. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
- if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not give any information to the user on the fact that the terminal is sending a short message;

- if the alpha identifier is not provided by the UICC, the terminal may give information to the user concerning what is happening.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

If the terminal is capable of SMS-MO, then it shall send the data as a Short Message TPDU to the destination address. The terminal shall give the result to the UICC using TERMINAL RESPONSE (indicating successful or unsuccessful transmission of the Short Message) after receiving a confirmation of the transfer from the network. If an alpha identifier was provided by the UICC, the terminal should not give any information to the user at the reception of SMS RP-ACK or RP-ERROR.

If the Short Message TPDU is unsuccessfully received by the network, the terminal shall inform the UICC using TERMINAL RESPONSE (network currently unable to process command). If a null alpha identifier was provided by the UICC, the terminal should not give any information to the user at the unsuccessful network reception.

The destination address and the SMSC address included in the SEND SHORT MESSAGE proactive command shall not be checked against those of the FDN list, even if the Fixed Dialling Number service is enabled.

#### 6.4.11 Void

#### 6.4.12 Void

#### 6.4.13 SET UP CALL

For a terminal of type ND, NK or NS the support of this command is optional.

This command is issued by the UICC to request a call set up.

Three types are defined:

- set up a call, but only if not currently busy on another call;
- set up a call, putting all other calls (if any) on hold;
- set up a call, disconnecting all other calls (if any) first.

For each of these types, the UICC may request the use of an automatic redial mechanism. The UICC may also request an optional maximum duration for the redial mechanism. The terminal shall attempt at least one call set-up.

In addition to the called party number, the command may contain capability configuration parameters (giving the bearer capability to request for the call) and the called party subaddress. The terminal shall use these in its call set-up request to the network. If the Bearer Capabilities element indicates that a Multi-media call is to be setup then the Terminal shall launch and use the relevant client to make the call (if class "h" is supported). The command may also include DTMF digits, which the terminal shall send to the network after the call has connected. The terminal shall not locally generate audible DTMF tones and play them to the user.

NOTE: On the downlink audio, DTMF tones reflected by the network may be heard.

It is possible for the UICC to request the terminal to set up an emergency call by supplying the number "112" as called party number. The terminal may translate this number in the appropriate technology specific number or procedure.

The number included in the SET UP CALL proactive command shall not be checked against those of the FDN list, even if the Fixed Dialling Number service is enabled.

Upon receiving this command, the terminal shall decide if it is able to execute the command. Examples are given below, but the list is not exhaustive:

- if the command is rejected because the terminal is busy on another call, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - currently busy on call).

If the terminal is able to set up the call on the serving network, the terminal shall:

- alert the user (as for an incoming call). This is the confirmation phase:
  - if the terminal does not have the technical means to alert the user (e.g., a terminal of type ND) or to receive confirmation from the user (e.g. a terminal of type NK), it may set up the call without explicit confirmation by the user.
- optionally, the UICC may include in this command one or two alpha-identifiers. The use of these alpha-identifiers by the terminal is described below:
  - if the first alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it during the user confirmation phase. This is also an indication that the terminal should not give any other information to the user during the user confirmation phase. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
  - if the first alpha identifier is not provided by the UICC or is a null data object (i.e. length = '00' and no value part), the terminal may give information to the user;
  - if the second alpha identifier (i.e. the one after the mandatory address object) is provided by the UICC and is not a null data object, the terminal shall use it during the call set-up phase and during the call. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
  - if the second alpha identifier is not provided by the UICC or is a null data object (i.e. length = '00' and no value part), the terminal may give information to the user;
  - if a terminal of type ND ignores any alpha identifier provided together with this command, due to lack of suitable alternative mechanism, the terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.
- if the call is accepted, the terminal shall then set up a call to the destination address given in the response data, with the relevant capability configuration parameters and called party subaddress (if provided by the UICC);
- if the call is not accepted, or rejected, then the terminal informs the UICC using TERMINAL RESPONSE (user did not accept the proactive command). The operation is aborted;
- if the user has indicated the need to end the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with "Proactive UICC session terminated by the user" result value;
- optionally, during call set-up, the terminal can give some audible or display indication concerning what is happening;
- once a CONNECT message has been received from the network (defined in TS 124 008 [20]), the terminal shall inform the UICC that the command has been successfully executed, using TERMINAL RESPONSE. Operation of the call then proceeds as normal.

If the first call set-up attempt is unsuccessful:

- if the UICC did not request redial then the terminal shall inform the UICC using TERMINAL RESPONSE (network currently unable to process command), and not redial to set-up the call;
- if the UICC requested redial, then the terminal may automatically redial the call (depending on its capability/configuration). In this case, the terminal shall not send a command result to the UICC concerning the first or any subsequent failed set-up attempts. If the call set-up has not been successful, and the terminal is not going to perform any more redials, or the time elapsed since the first call set-up attempt has exceeded the duration requested by the UICC, then the terminal shall inform the UICC using TERMINAL RESPONSE (network currently unable to process command), and the redial mechanism shall be terminated;

- if the user stops the call set-up attempt or the redial mechanism before a result is received from the network, the terminal informs the UICC using TERMINAL RESPONSE (user cleared down call before connection or network release).

If the terminal supports the storage of call set-up details in the UICC, the terminal shall not store in the UICC the call set-up details (called party number and associated parameters) sent by the UICC in this command.

#### 6.4.14 POLLING OFF

This command disables the Proactive Polling (defined in TS 102 221 [1] and in TS 151 011 [8]). UICC Presence Detection (defined in TS 102 221 [1]) or SIM presence detection (defined in TS 151 011 [8]) are not affected by this command.

#### 6.4.15 PROVIDE LOCAL INFORMATION

This command requests the terminal to send current local information to the UICC. At present, this information is restricted to:

- location information;
- the terminal identity (e.g. IMEI, IMEISV, ESN, MEID) of the terminal;
- the network measurement results;
- the current date, time and time zone;
- the current terminal language setting;
- the timing advance (access network dependent);
- the charge state of the battery (if class "g" is supported);
- the current access technology or technologies;
- the current network search mode;
- the Broadcast Network information (if class "o" is supported).

The terminal shall return the requested local information within a TERMINAL RESPONSE. The terminal shall return the current date and time as set by the user. If available, the terminal shall also return the time zone known from the network. If the time zone information is not available, the terminal shall return 'FF' for this element.

If language setting is requested, the terminal shall return the currently used language.

If the access technology is requested, the terminal shall return the current access technology or technologies that the terminal is using.

#### 6.4.16 SET UP EVENT LIST

The UICC shall use this command to supply a set of events. This set of events shall become the current list of events for which the terminal is to monitor.

Any subsequent SET UP EVENT LIST command replaces the current list of events supplied in the previous SET UP EVENT LIST command. The SET UP EVENT LIST command can also be used to remove the entire list of events current in the terminal; see clause 6.6.16. The list of events provided by the UICC in the last SET UP EVENT LIST command shall be removed if the terminal is powered off or the UICC is removed or a reset is performed.

When the terminal has successfully accepted or removed the list of events, it shall send TERMINAL RESPONSE (OK) to the UICC.

When the terminal is not able to successfully accept or remove the list of events, it shall send TERMINAL RESPONSE (Command beyond terminal's capabilities).



When one of the events in the current list occurs, then the terminal shall use the Event Download mechanism to transfer details of the event to the UICC, see clause 7.5.

## 6.4.17 PERFORM CARD APDU

This clause applies if class "a" is supported.

This command requests the terminal to send an APDU command to the additional card (Card x).

The command includes:

- the additional card reader identifier, which is part of the Device Identities object;
- the APDU command to be performed.

Upon receiving this command, the terminal shall decide if it is able to execute the command:

- if the command is rejected because the card reader identity is not valid, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card reader not valid);
- if the command is rejected because the card reader is not present or has been removed, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card reader removed or not present);
- if the command is rejected because the card is not present or has been removed, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card removed or not present);
- if the command is rejected because the card reader is busy, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card reader busy);
- if the command is rejected because the card is not powered on, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card powered off);
- if the command is rejected because the received C-APDU format is not valid, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - C-APDU format error).

If the terminal is able to transfer the C-APDU to the addressed card, the terminal shall:

- transfer the C-APDU to the addressed card, through the selected terminal - Card x protocol;
- extract the R-APDU data from the addressed card if so requested by the UICC;
- if the command fails because no response is received from Card x, inform the UICC using TERMINAL RESPONSE (MultipleCard command error - Card mute);
- if the command fails because of any form of transmission error, inform the UICC using TERMINAL RESPONSE (MultipleCard command error - Transmission error);
- if the command fails because the terminal does not support the protocol used by Card x, inform the UICC using TERMINAL RESPONSE (MultipleCard command error - Protocol not supported);
- if the command is performed successfully from a protocol point of view, include the R-APDU within the TERMINAL RESPONSE command.

## 6.4.18 POWER OFF CARD

This clause applies if class "a" is supported.

This command requests the terminal to close a session with the additional card (Card x).

The command includes the additional card reader identifier, which is part of the Device Identities object.

Upon receiving this command, the terminal shall decide if it is able to execute the command:

- if the command is rejected because the card reader identity is not valid, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card reader not valid);

- if the command is rejected because the card reader is not present or has been removed, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card reader removed or not present);
- if the command is rejected because the card is not present or has been removed, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card removed or not present);
- if the command is rejected because the card reader is busy, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card reader busy).

If the terminal is able to execute the command, the addressed Card x shall be deactivated according to ISO/IEC 7816-3 [13].

### 6.4.19 POWER ON CARD

This clause applies if class "a" is supported.

This command requests the terminal to start a session with the additional card (Card x).

The command includes the additional card reader identifier, which is part of the Device Identities object.

Upon receiving this command, the terminal shall decide if it is able to execute the command:

- if the command is rejected because the card reader identity is not valid, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card reader not valid);
- if the command is rejected because the card reader is not present or has been removed, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card reader removed or not present);
- if the command is rejected because the card is not present or has been removed, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card removed or not present);
- if the command is rejected because the card reader is busy, the terminal informs the UICC using TERMINAL RESPONSE (MultipleCard command error - Card reader busy).

If the terminal is able to execute the command, and the addressed Card x is powered off, the terminal shall activate the addressed Card x according to ISO/IEC 7816-3 [13]. If the addressed Card x is already powered on, the terminal shall treat the POWER ON CARD command as a warm reset, as defined in ISO/IEC 7816-3 [13].

The terminal shall return the Answer To Reset within the TERMINAL RESPONSE command. If no ATR is received, the terminal shall inform the UICC using TERMINAL RESPONSE (MultipleCard command error - Card mute).

Application writers are advised that the Card x should not be powered up for longer than necessary due to battery life considerations.

### 6.4.20 GET READER STATUS

This clause applies if class "a" is supported.

This command requests the terminal to get information about all interfaces or the indicated interface to additional card reader(s). This information is restricted to:

- card reader status;
- card reader identifier.

The terminal shall return the requested information from the interfaces to additional card reader(s) within a TERMINAL RESPONSE command.

## 6.4.21 TIMER MANAGEMENT

This command requests the terminal to manage timers running physically in the terminal. The possible actions on timers are defined below:

- start a timer;
- deactivate a timer;
- get the current value of a timer.

The UICC and the terminal are able to manage eight different timers running in parallel. The possible duration of a timer is between 1 s and 24 h. The resolution of a timer is 1 second. The precision of the returned value cannot be relied upon in all cases due to potential terminal activities. When the terminal is switched off or a reset is performed, all timers are deactivated in the terminal.

For a given timer:

- when the UICC requests the terminal to start the timer with a duration, then:
  - the terminal shall start the timer with the duration given by the UICC, even if this timer is already running. When a timer is started, it takes the value given by the UICC, and is then decremented. The terminal shall inform the UICC that the command has been successfully executed, using **TERMINAL RESPONSE (OK)**.
- when the UICC requests the terminal to deactivate the timer, then:
  - if the timer is running, the terminal shall deactivate the timer. This prevents the UICC from receiving unnecessary information at the expiration of a timer. The terminal shall pass the current value of the timer (i.e. the duration that remains before the timer elapses) to the UICC, using **TERMINAL RESPONSE**;
  - if the timer is already deactivated, the terminal shall inform the UICC using **TERMINAL RESPONSE ("action in contradiction with the current timer state")**.
- when the UICC requests the terminal to get the current value of the timer, then:
  - if the timer is running, the terminal shall pass the current value of the timer (i.e. the duration that remains before the timer elapses) to the UICC, using **TERMINAL RESPONSE**;
  - if the timer is deactivated, the terminal shall inform the UICC using **TERMINAL RESPONSE ("action in contradiction with the current timer state")**.

When a timer expires (i.e. reaches zero), the terminal shall use the Timer Expiration mechanism to transfer the identifier of the timer that has expired and the difference between the time when this transfer occurs and the time when the timer was initially started. The terminal shall then deactivate the timer.

## 6.4.22 SET UP IDLE MODE TEXT

The UICC shall supply a text string, which shall be displayed by the terminal as an idle mode text if the terminal is able to do it. The presentation style is left as an implementation decision to the terminal manufacturer. The idle mode text shall be displayed in a manner that ensures that neither the network name nor the service providers name are affected.

If idle mode text is competing with other information to be displayed on the same area, for instance a CB message, the idle mode text shall be replaced by the other information. It is up to the terminal to restore the idle mode text when the other information has no longer to be displayed.

The text shall be removed from the terminal's memory and display if either:

- the terminal is powered off; or
- the UICC is removed or a reset is performed; or
- a **REFRESH** command occurs with "initialization" or "reset".

Any subsequent SET UP IDLE MODE TEXT command replaces the current idle mode text of the previous SET-UP IDLE MODE TEXT. The SET UP IDLE MODE TEXT command can also be used to remove an idle mode text from the terminal, see clause 6.6.22.

When the terminal has successfully integrated or removed an idle mode text, it shall send TERMINAL RESPONSE (OK) to the UICC.

When the terminal is not able to successfully integrate or remove the idle mode text, it shall send TERMINAL RESPONSE "Command beyond terminal's capabilities" to the UICC.

### 6.4.23 RUN AT COMMAND

This clause applies if class "b" is supported by the terminal and enabled by the subscriber through the terminal.

If this feature is enabled, the UICC uses this command to send an AT Command to the terminal as though initiated by an attached TE. The terminal shall then return an AT Response within a TERMINAL RESPONSE to the UICC.

If this feature is disabled in the card or in the terminal, or the terminal does not support the RUN AT COMMAND, then if the CAT receives an instruction from the network to issue the command, the CAT should return an error to the network.

Optionally, the UICC may include in this command an alpha identifier. The use of this alpha identifier by the terminal is described below:

- if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it to inform the user. This is also an indication that the terminal should not give any other information to the user on the fact that the terminal is performing an AT command. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
- if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not give any information to the user on the fact that the terminal is performing an AT command;
- if the alpha identifier is not provided by the UICC, the terminal may give information to the user concerning what is happening.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

### 6.4.24 SEND DTMF

For a terminal of type NS the support of this command is optional.

This command requests the terminal to send a DTMF string after a call has been successfully established either by the proactive command SET UP CALL or the user. This command is independent of sending DTMF within the call set up (as defined in the SET UP CALL command) and therefore, can be used at any time during a call.

The terminal shall not locally generate audible DTMF tones and play them to the user.

NOTE: On the downlink audio, DTMF tones reflected by the network may be heard.

It shall be possible for the user to deactivate this command.

The sending of a DTMF string applies only to the currently active call.

The TERMINAL RESPONSE indicating that the command has been performed successfully shall be sent after the complete DTMF string has been sent to the network by the terminal.

If the command is sent in idle mode, or a call is terminated or put on hold before the complete DTMF string has been sent to the network, the terminal shall inform the UICC using TERMINAL RESPONSE '20' with the additional information "Not in speech call".

If the user indicates the need to end the proactive UICC session whilst the terminal is sending the DTMF string, the terminal shall stop sending the DTMF string and shall send a TERMINAL RESPONSE with "Proactive UICC session terminated by the user" result value.

Optionally, the UICC may include in this command an alpha identifier. The use of this alpha identifier by the terminal is described below:

- if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it to inform the user. This is also an indication that the terminal should not give any other information to the user on the fact that the terminal is performing a SEND DTMF command. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
- if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not give any information to the user on the fact that the terminal is performing a SEND DTMF command;
- if the alpha identifier is not provided by the UICC, the terminal may give information to the user concerning what is happening.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

## 6.4.25 LANGUAGE NOTIFICATION

For a terminal of type NL the support of this command is optional.

The UICC shall use this command to notify the terminal about the language currently used for any text string within proactive commands or envelope command responses.

The notified language stays valid within the terminal until the end of the card session or upon executing another LANGUAGE NOTIFICATION command.

When the CAT application is not aware of the current CAT application language, no specific language is in use or several languages are in use, the UICC may notify non-specific language. This has the effect of cancelling a previous specific LANGUAGE NOTIFICATION.

Two types of language notification are defined:

- specific, where an additional Language object shall be included by the UICC;
- non-specific, where no Language object shall be included by the UICC.

Regardless of whether the terminal recognizes the notified language or not, the terminal shall send TERMINAL RESPONSE (OK) to the UICC.

The terminal may use the language included in LANGUAGE NOTIFICATION as appropriate. For instance, this could be done to avoid a mix of languages in screen displays combining terminal MMI and CAT originating text strings.

## 6.4.26 LAUNCH BROWSER

For a terminal of type ND or NK the support of this command is optional.

This command is used to request a browser inside a browser-enabled terminal to interpret the content corresponding to a URL.

Upon receiving this command, the terminal shall decide if it is able to execute the command. Examples are given below, but the list is not exhaustive:

- if the command is rejected because the browser on the terminal is busy or not available, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - browser unavailable);

- if the command is rejected because the bearer provided in the command is not available, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - bearer unavailable).

If the terminal is able to execute the command:

- the terminal shall inform the UICC that the command has been successfully taken into account, using TERMINAL RESPONSE;
- the UICC shall end the proactive session;
- then the terminal shall request content using the URL;
- if an error occurs when accessing the resource indicated in the URL, the terminal shall send to the UICC a browsing status event reporting the error (if the browsing status event is part of the event list).

If the gateway addresses and/or the bearer objects are present in the command and are non null data objects, then the browser shall use these data to request content using the URL. If the gateway addresses, bearer objects, Provisioning File Reference, Browser Identity or URL are null objects or missing, then the terminal shall use default values (for an example, see reference in annex J).

The terminal shall ask the user for confirmation using the Alpha Identifier/Icon Identifier (user confirmation phase) if present, when it receives a LAUNCH BROWSER command which requests the existing browser session connected to a new URL or to terminate a browser session.

The way the terminal requests content using the URL is outside the scope of the present document (for an example, see reference in annex J).

NOTE: There is a maximum size for the URL that can be given in argument of this proactive command.

## 6.4.27 OPEN CHANNEL

### 6.4.27.1 OPEN CHANNEL related to CS bearer

This clause applies if class "e" is supported.

Upon receiving this command, the terminal shall decide if it is able to execute the command. The UICC shall indicate whether the terminal should establish the link immediately or upon receiving the first transmitted data (on demand).

The UICC provides to the terminal a list of parameters necessary to establish a link.

The UICC may request the use of an automatic reconnection mechanism. The UICC may also request an optional maximum duration for the reconnection mechanism. The terminal shall attempt at least one link establishment set-up.

The UICC may also request an optional maximum duration for the terminal to automatically release the link if no data is exchanged.

If the Fixed Dialling Number service is enabled, the address included in the OPEN CHANNEL proactive command shall not be checked against those of the FDN list.

Upon receiving this command, the terminal shall decide if it is able to execute the command. Examples are given below, but the list is not exhaustive:

- if immediate link establishment is requested and the terminal is unable to set-up a channel using the exact parameters provided by the UICC, the terminal sets up the channel using the best parameters it can support and informs the UICC of the Channel status and the modified parameters using TERMINAL RESPONSE (Command performed with modification);
- if immediate link establishment is requested and the terminal is unable to set-up the link with the network using the exact parameters provided by the UICC, the terminal informs the UICC using TERMINAL RESPONSE (Network currently unable to process command). The operation is aborted;

- if on demand link establishment is requested and the terminal is unable to set-up a channel using the exact parameters provided by the UICC, the terminal sets up the channel using the best parameters it can support and informs the UICC of the Channel status and the modified parameters using TERMINAL RESPONSE (Command performed with modification);
- if the command is rejected because the terminal has no channel left with the requested bearer capabilities, the terminal informs the UICC using TERMINAL RESPONSE (Bearer independent protocol error). The operation is aborted;
- if the user does not accept the channel set-up, the terminal informs the UICC using TERMINAL RESPONSE (User did not accept the proactive command). The operation is aborted;
- if the user has indicated the need to end the proactive UICC session, the terminal informs the UICC using TERMINAL RESPONSE (Proactive UICC session terminated by the user). The operation is aborted;
- if the command is rejected because the terminal is busy on another call, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - currently busy on call). The operation is aborted.

The terminal shall inform the UICC that the command has been successfully executed using TERMINAL RESPONSE:

- if immediate link establishment is requested, the terminal allocates buffers, sets up the link and informs the UICC and reports the Channel status using TERMINAL RESPONSE (Command performed successfully);
- if on demand link establishment is requested, the terminal allocates buffers, informs the UICC and reports the Channel status using TERMINAL RESPONSE (Command performed successfully).

If the terminal is able to set up the channel on the serving network, the terminal shall:

- alert the user (as for an incoming call). This is the confirmation phase;

A terminal of type NK or type ND may not alert the user and may open the channel without explicit confirmation by the user.

- optionally, the UICC may include in this command an alpha-identifier. The use of this alpha-identifier by the terminal is described below:
  - if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it during the user confirmation phase. This is also an indication that the terminal should not give any other information to the user during the user confirmation phase. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
  - if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not give any information to the user or ask for user confirmation;
  - if the alpha identifier is not provided by the UICC, the terminal may give information to the user;

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

- if the user accepts the channel, the terminal shall then set up a channel;
- if the user does not accept the channel or rejects the channel, then the terminal informs the UICC using TERMINAL RESPONSE (user did not accept the proactive command). The operation is aborted;
- if the user has indicated the need to end the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with (Proactive UICC session terminated by the user) result value;
- optionally, during call set-up, the terminal can give some audible or display indication concerning what is happening.

If the first link set-up attempt is unsuccessful:

- if the UICC did not request link re-connection then the terminal shall inform the UICC using **TERMINAL RESPONSE** (network currently unable to process command), and not retry to set-up the link:
  - if the UICC requested link re-connection, then the terminal may automatically retry to set-up the link (depending on its configuration capabilities). In this case, the terminal shall not send a command result to the UICC concerning the first or any subsequent failed set-up attempts. If the link set-up has not been successful, and the terminal is not going to perform any more re-tries, or the time elapsed since the first link set-up attempt has exceeded the duration requested by the UICC, then the terminal shall inform the UICC using **TERMINAL RESPONSE** (network currently unable to process command), and the re-try mechanism shall be terminated;
  - if the user stops the link set-up attempt or the re-try mechanism before a result is received from the network, the terminal informs the UICC using **TERMINAL RESPONSE** (user cleared down call before connection or network release).

If the terminal supports the storage of call set-up details in the UICC, the terminal shall not store in the UICC the channel set-up details (called party number and associated parameters) sent by the UICC in this command.

#### 6.4.27.2 OPEN CHANNEL related to packet data service bearer

This clause applies if class "e" is supported.

Upon receiving this command, the terminal shall decide if it is able to execute the command. The UICC shall indicate whether the terminal should establish the link immediately, in background mode or upon receiving the first transmitted data (on demand).

The UICC provides to the terminal a list of parameters necessary to activate a packet data service.

The terminal shall attempt at least one packet data service activation.

Upon receiving this command, the terminal shall decide if it is able to execute the command. Examples are given below, but the list is not exhaustive:

- if immediate or background packet data service activation is requested and the terminal is unable to set-up a channel using the exact parameters provided by the UICC, the terminal sets up the channel using the best parameters it can support and informs the UICC of the Channel status and the modified parameters using **TERMINAL RESPONSE** (Command performed with modification);
- if immediate packet data service activation is requested and the terminal is unable to activate the packet data service with the network using the exact parameters provided by the UICC, the terminal informs the UICC using **TERMINAL RESPONSE** (Network currently unable to process command). The operation is aborted;
- if background mode packet data service activation is requested and the terminal is unable to activate the packet data service with the network using the exact parameters provided by the UICC, the terminal informs the UICC using a channel status event (link not established - no further info). The operation is aborted;
- if on demand link establishment is requested and the terminal is unable to set-up a channel using the exact parameters provided by the UICC, the terminal sets up the channel using the best parameters it can support and informs the UICC of the Channel status and the modified parameters using **TERMINAL RESPONSE** (Command performed with modification);
- if the command is rejected because the terminal has no channel left with the requested bearer capabilities, the terminal informs the UICC using **TERMINAL RESPONSE** (Bearer independent protocol error). The operation is aborted;
- if the user does not accept the channel set-up, the terminal informs the UICC using **TERMINAL RESPONSE** (User did not accept the proactive command). The operation is aborted;
- if the user has indicated the need to end the proactive UICC session, the terminal informs the UICC using **TERMINAL RESPONSE** (Proactive UICC session terminated by the user). The operation is aborted;



- if the command is rejected because the class B terminal is busy on a call, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - currently busy on call). The operation is aborted;
- if background mode packet data service activation is requested, the terminal allocates buffers, starts activation of packet data service, informs the UICC and reports the Channel status immediately using TERMINAL RESPONSE (Command performed successfully). At the end of activation, the terminal shall send a channel status event (link established or link not established - no further info).

The terminal shall inform the UICC that the command has been successfully executed using TERMINAL RESPONSE:

- if immediate Packet data service activation is requested, the terminal allocates buffers, activates the packet data service and informs the UICC and reports the Channel status using TERMINAL RESPONSE (Command performed successfully);
- if on demand packet data service activation is requested, the terminal allocates buffers, informs the UICC and reports the Channel status using TERMINAL RESPONSE (Command performed successfully).

If the terminal is able to set up the channel on the serving network, the terminal shall then enter the confirmation phase described hereafter; optionally, the UICC may include in this command an alpha-identifier. The use of this alpha-identifier by the terminal is described below:

- if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it during the user confirmation phase. This is also an indication that the terminal should not give any other information to the user during the user confirmation phase. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
- if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not give any information to the user or ask for user confirmation;
- if the alpha identifier is not provided by the UICC, the terminal may give information to the user;
- if the user does not reject the channel, the terminal shall then set up a channel;
- if the user does not accept the channel or rejects the channel, then the terminal informs the UICC using TERMINAL RESPONSE (user did not accept the proactive command). The operation is aborted;
- if the user has indicated the need to end the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with (Proactive UICC session terminated by the user) result value;
- optionally, during packet data service activation, the terminal can give some audible or display indication concerning what is happening;
- if the user stops the packet data service activation attempt before a result is received from the network, the terminal informs the UICC using TERMINAL RESPONSE (user cleared down call before connection or network release).

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

A terminal of type NK or type ND may not alert the user and may open the channel without explicit confirmation by the user.

### 6.4.27.3 OPEN CHANNEL related to local bearer

This clause applies if classes "e" and "f" are supported.

This command is used to establish a connection using a local bearer (Bluetooth, IrDA, RS232, USB). The UICC can act as a server or a client. In the server use case, the UICC performs an OPEN CHANNEL only after having received a Local Connection event from the terminal.

Upon receiving this command, the terminal shall decide if it is able to execute the command. The UICC shall indicate whether the terminal should establish the link immediately or upon receiving the first transmitted data (on demand).

The UICC provides to the terminal a list of parameters necessary to establish a link.

The UICC may request the use of an automatic reconnection mechanism. The UICC may also request an optional maximum duration for the reconnection mechanism. The terminal shall attempt at least one link establishment set-up.

The UICC may also request an optional maximum duration for the terminal to automatically release the link if no data is exchanged.

Upon receiving this command, the terminal shall decide if it is able to execute the command. Examples are given below, but the list is not exhaustive:

- if immediate link establishment is requested and the terminal is unable to set-up a channel using the exact parameters provided by the UICC, the terminal sets up the channel using the best parameters it can support and informs the UICC of the Channel status and the modified parameters using TERMINAL RESPONSE (Command performed with modification);
- if immediate link establishment is requested and the terminal is unable to set-up the link with the network using the exact parameters provided by the UICC, the terminal informs the UICC using TERMINAL RESPONSE (Network currently unable to process command). The operation is aborted;
- if on demand link establishment is requested and the terminal is unable to set-up a channel using the exact parameters provided by the UICC, the terminal sets up the channel using the best parameters it can support and informs the UICC of the Channel status and the modified parameters using TERMINAL RESPONSE (Command performed with modification);
- if the command is rejected because the terminal has no channel left with the requested bearer capabilities, the terminal informs the UICC using TERMINAL RESPONSE (Bearer independent protocol error). The operation is aborted;
- if the user does not accept the channel set-up, the terminal informs the UICC using TERMINAL RESPONSE (User did not accept the proactive command). The operation is aborted;
- if the user has indicated the need to end the proactive UICC session, the terminal informs the UICC using TERMINAL RESPONSE (Proactive UICC session terminated by the user). The operation is aborted;
- if the command is rejected because the terminal is busy on another call, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - currently busy on call). The operation is aborted.

The terminal shall inform the UICC that the command has been successfully executed using TERMINAL RESPONSE:

- if immediate link establishment is requested, the terminal allocates buffers, sets up the link and informs the UICC and reports the Channel status using TERMINAL RESPONSE (Command performed successfully);
- if on demand link establishment is requested, the terminal allocates buffers, informs the UICC and reports the Channel status using TERMINAL RESPONSE (Command performed successfully).

If the terminal is able to set up the channel on the requested local bearer, the terminal shall:

- alert the user (as for an incoming call). This is the confirmation phase;

A terminal of type NK or type ND may not alert the user and may open the channel without explicit confirmation by the user.

- optionally, the UICC may include in this command an alpha-identifier. The use of this alpha-identifier by the terminal is described below:
  - if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it during the user confirmation phase. This is also an indication that the terminal should not give any other information to the user during the user confirmation phase. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);

- if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not give any information to the user or ask for user confirmation;
- if the alpha identifier is not provided by the UICC, the terminal may give information to the user;

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

- if the user accepts the channel, the terminal shall then set up a channel;
- if the user does not accept the channel or rejects the channel, then the terminal informs the UICC using TERMINAL RESPONSE (user did not accept the proactive command). The operation is aborted;
- if the user has indicated the need to end the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with (Proactive UICC session terminated by the user) result value;
- optionally, during call set-up, the terminal can give some audible or display indication concerning what is happening;
- if the first link set-up attempt is unsuccessful;
- if the UICC did not request link re-connection then the terminal shall inform the UICC using TERMINAL RESPONSE (network currently unable to process command), and not retry to set-up the link:
  - if the UICC requested link re-connection, then the terminal may automatically retry to set-up the link (depending on its configuration capabilities). In this case, the terminal shall not send a command result to the UICC concerning the first or any subsequent failed set-up attempts. If the link set-up has not been successful, and the terminal is not going to perform any more re-tries, or the time elapsed since the first link set-up attempt has exceeded the duration requested by the UICC, then the terminal shall inform the UICC using TERMINAL RESPONSE (network currently unable to process command), and the re-try mechanism shall be terminated;
  - if the user stops the link set-up attempt or the re-try mechanism before a result is received from the network, the terminal informs the UICC using TERMINAL RESPONSE (user cleared down call before connection or network release).

#### 6.4.27.4 OPEN CHANNEL related to Default (network) Bearer

This clause applies only if class "e" is supported.

Upon receiving this command, the terminal shall decide if it is able to execute the command. The UICC shall indicate whether the terminal should establish the link immediately or upon receiving the first transmitted data (on demand).

The terminal is responsible for providing the parameters necessary to establish the connection (e.g. APN for GPRS, Address for CSD, etc.).

Upon receiving this command, the terminal shall decide if it is able to execute the command. Example behaviours are listed in clauses for the selected bearer.

The terminal shall inform the UICC that the command has been successfully executed using TERMINAL RESPONSE:

- If immediate connection is requested (link establishment or PDP context activation), the terminal allocates buffers, sets up the link or activates the PDP context (depending of the kind of connection), and informs the UICC and reports the Channel status using TERMINAL RESPONSE (Command performed successfully).
- If on demand connection is requested (link establishment or PDP context activation), the terminal allocates buffers, informs the UICC and reports the Channel status using TERMINAL RESPONSE (Command performed successfully).

- If the terminal is able to set up the channel on the serving network, the terminal shall follow the different actions of the chosen bearer (see appropriate clauses).

A terminal of type NK or type ND may not alert the user and may open the channel without explicit confirmation by the user.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

#### 6.4.27.5 OPEN CHANNEL related to UICC Server Mode

This clause applies only if class "e" is supported.

The UICC acts as TCP server for a client application (e.g. web browser) in the terminal. The terminal shall listen on the localhost IP address (e.g. 127.0.0.1 for IPv4) at the TCP port given in the command and forward incoming/outgoing data on this port to/from the UICC.

Upon receiving this command, the terminal shall decide if it is able to execute the command. If the terminal is able to execute the command:

- the terminal shall listen on the indicated TCP port;
- it shall inform the UICC that the command has been successfully executed using TERMINAL RESPONSE with connection status set to "TCP in LISTEN state"; and
- optionally, the UICC may include in the command an alpha identifier. If the alpha identifier is provided by the UICC and is not a null data object, the terminal may use it to inform the user. User confirmation is not required for UICC Server Mode.

As soon as a client successfully establishes a connection to the TCP port, the terminal shall inform the UICC about this event by sending a channel status event with connection status set to "TCP in ESTABLISHED state".

Only one TCP connection can be handled on one BIP channel at any point of time. If a second connection in parallel is needed, the UICC shall open a second BIP channel on the same or a different port.

**NOTE:** Several BIP channels may be associated to the same server port in LISTEN state. The terminal may choose any of these, if a connection request comes in from a client.

If a TCP disconnect occurs (i.e. client closed the connection) while the BIP connection is still open, the terminal:

- shall be ready to accept a new TCP connection on the server port of the BIP connection; and
- shall discard current buffered data and inform the UICC about the new connection status "TCP in LISTEN state".

If a TCP disconnect occurs while the BIP connection is still open, and the terminal is not able to fall back to the connection status "TCP in LISTEN state" because of a lack of resources, then the terminal shall inform the UICC about the new connection status "TCP in CLOSED state".

If the terminal is unable to process the command (the list is not exhaustive):

- if the command is rejected because the terminal has reached its maximum TCP server connection capabilities, the terminal informs the UICC using TERMINAL RESPONSE (Command beyond terminal's capabilities);
- if the command is rejected because the requested TCP port is not available in the terminal, the terminal informs the UICC using TERMINAL RESPONSE (Bearer Independent Protocol error, port not available).

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

#### 6.4.27.6 OPEN CHANNEL related to Terminal Server Mode

This clause applies only if class "e" and class "k" are supported.

For a terminal using TCP or UDP to connect terminal applications, the following applies: the UICC can open a connection to a server in the terminal. If a TCP connection is requested, the terminal assigns an available source port and issues an active OPEN request (as defined in RFC 793 [10]) to the port number given in the command at the localhost IP address (e.g. 127.0.0.1 for IPv4). If UDP connection is requested, the terminal assigns an available source port and will send future data as datagrams (as defined in RFC 768 [9]) to the port number given in the command at the localhost IP address (e.g. 127.0.0.1 for IPv4). In both cases the terminal forwards incoming/outgoing data with the allocated pair of source and destination ports to/from the UICC. Dependent on the terminal implementation, opening a connection to a port may cause a terminal application (which does not need additional launch parameters) to be launched.

If the UICC wants to provide additional launch parameters for a terminal application to be launched, this shall be done as follows:

- The UICC shall indicate "launch parameters following" in the OPEN CHANNEL command.
- The UICC shall send one or several SEND DATA commands containing the launch parameters. The last or only command containing the launch parameters shall indicate "send immediate".
- The terminal shall launch the application only after having received all launch parameters and report the result of the application launch to the UICC in the TERMINAL RESPONSE to the last SEND DATA command.

A terminal not using TCP or UDP to connect terminal applications may also use this mechanism for opening a direct communication channel between the UICC and a terminal application. In that case, the port number is used as a simple reference for the application. Once the channel is open, the terminal shall forward incoming/outgoing data to/from the UICC directly to the application.

Closing a channel (using the Close Channel command) shall not close terminal applications launched by opening the channel in Terminal Server Mode. The Close Channel command shall only close the communication channel between the UICC and the application.

- Upon receiving this command, the terminal shall decide if it is able to execute the command.

Only one TCP connection or direct communication channel can be handled on one BIP channel at any point of time. If a second connection in parallel is needed, the UICC shall open a second BIP channel on the same or a different port.

If a TCP disconnect occurs or the direct communication channel is closed by the terminal while the BIP connection is still open, the terminal:

- shall inform the UICC using a channel status event (TCP in CLOSED state/direct communication channel closed), and wait for a CLOSE CHANNEL command from the UICC.

If the terminal is unable to process the command (the list is not exhaustive):

- if the command is rejected because the terminal has reached its maximum connection capabilities, the terminal informs the UICC using TERMINAL RESPONSE (Command beyond terminal's capabilities);
- if the command is rejected because the requested port is not available in the terminal, the terminal informs the UICC using TERMINAL RESPONSE (Bearer Independent Protocol error, port not available);
- if the command is rejected because no server is listening at the requested port or no application has registered for a direct communication channel associated to the port number, the terminal informs the UICC using TERMINAL RESPONSE (Requested UICC/terminal interface transport level not available);
- if the terminal has information about the launch parameters required by an application to be launched and it could not launch the application due to missing or unusable launch parameters, the command is rejected and the terminal informs the UICC using TERMINAL RESPONSE (Bearer Independent protocol error, launch parameters missing or incorrect).

For an application to be launched without additional launch parameters, the following applies: if the terminal is able to start the application, the terminal shall then enter the confirmation phase described hereafter; optionally, the UICC may include in this command an alpha-identifier. The use of this alpha-identifier by the terminal is described below:

- if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it during the user confirmation phase. This is also an indication that the terminal shall not give any other information to the user during the user confirmation phase. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
- if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal shall not give any information to the user or ask for user confirmation;
- if the alpha identifier is not provided by the UICC, the terminal may give information to the user;
- if the user does not reject the channel, the terminal shall then start the application;
- if the user does not accept the channel or rejects the channel, then the terminal informs the UICC using TERMINAL RESPONSE (user did not accept the proactive command). The operation is aborted;
- if the user has indicated the need to end the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with (Proactive UICC session terminated by the user) result value.

For an application to be launched with additional launch parameters, alpha identifier, text attribute and icon identifier shall be ignored.

A terminal of type NK or type ND shall not indicate support of confirmation parameters for OPEN CHANNEL in Terminal Server Mode in its TERMINAL PROFILE.

## 6.4.28 CLOSE CHANNEL

This clause applies if class "e" is supported.

This command requests the terminal to close the channel corresponding to the Channel identifier as indicated in the Device identities.

Upon receiving this command, the terminal shall decide if it is able to execute the command:

- if the command is rejected because the channel identifier is not valid, the terminal informs the UICC using TERMINAL RESPONSE (Bearer independent protocol error);
- if the command is rejected because the requested channel is in error, the terminal informs the UICC using TERMINAL RESPONSE (Bearer independent protocol error).

If the terminal is able to process the command:

- the terminal shall release the data transfer, discard the remaining data and inform the UICC that the command has been successfully executed, using TERMINAL RESPONSE;
- optionally, during CLOSE CHANNEL, the terminal can give some audible or display indication concerning what is happening. In this intention, the UICC may include in this command an alpha-identifier. The use of this alpha-identifier by the terminal is described below:
  - if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it to indicate the link closing phase. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
  - if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not give any indication to the user during the link closing phase;
  - if the alpha identifier is not provided by the UICC, the terminal may give an indication to the user during the link closing phase.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

**For a connection in UICC Server Mode**, the UICC may have to close the TCP connection without closing the BIP channel. In that case, the UICC indicates in the command detail a request to the terminal to close the TCP connection and go back to "TCP in LISTEN state". Upon receiving this request, the terminal shall close the TCP connection and return to the "TCP in LISTEN state" without closing the BIP channel.

## 6.4.29 RECEIVE DATA

This clause applies if class "e" is supported.

This command requests the terminal to return data from a dedicated Channel identifier (indicated in the Device identities) according to the number of bytes specified by the UICC.

Upon receiving this command, the terminal shall return the data available in the Rx buffer corresponding to the Channel identifier. Examples are given below, but the list is not exhaustive.

If the terminal is unable to process the command:

- if the command is rejected because the requested channel is already closed the terminal informs the UICC using TERMINAL RESPONSE (Bearer independent protocol error);
- if the user has indicated the need to end the proactive UICC session, the terminal informs the UICC using TERMINAL RESPONSE (Proactive UICC session terminated by the user).

If the terminal is able to process the command:

- if the requested number of bytes is available in the buffer, the terminal shall inform the UICC that the command has been successfully executed, using TERMINAL RESPONSE and return the requested data and the number of bytes remaining in the channel buffer (or FF if more than the maximum bytes remains);
- if the requested number of bytes is available in the buffer but the whole requested data cannot be included in the TERMINAL RESPONSE because of APDU size limits, the terminal shall return the maximum number of bytes possible according to the length of other TLVs. The terminal shall inform the UICC that the command has been successfully executed, using TERMINAL RESPONSE and shall indicate the number of bytes remaining in the channel buffer (or FF if more than the maximum bytes remains);
- if the requested number of bytes is not yet available in the buffer, the terminal shall NOT wait for the requested number of bytes to arrive. The terminal shall inform the UICC, using TERMINAL RESPONSE (Command performed with missing information) and returns the data currently available in the channel buffer;
- in the case of packet/datagram transmission, the terminal shall put in the Rx buffer a complete packet SDU and only one at one time. For example, if UDP datagrams are received by the terminal, the latter shall insert only the SDU of each UDP packet received in the Rx buffer. After one SDU has been downloaded by the UICC (using one or several RECEIVE DATA commands), the terminal shall insert the next SDU of UDP datagram, and so on;
- optionally, the UICC may include in the command an alpha identifier. The use of this alpha identifier by the terminal is described below:
  - if the alpha identifier is provided by the UICC, the terminal shall use it to inform the user. The terminal may also use it to inform the user during data transfer. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
  - if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not inform the user during data transfer;
  - if the alpha identifier is not provided by the UICC, the terminal may inform the user during data transfer.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

### 6.4.30 SEND DATA

This clause applies if class "e" is supported.

This command requests the terminal to send data through a previously set up data channel corresponding to a dedicated Channel identifier (indicated in the Device identities). The UICC informs the terminal if the data is:

- to be sent immediately;
- or to be stored in a Tx buffer. Then it is up to the terminal to manage the data sending in order to use the bearer in an optimized way. To send the data stored in a Tx buffer, the terminal shall be notified by a "send data immediately" and it shall consider the data presently and previously concatenated in its Tx buffer as one SDU, and send it in only one PDU. The Tx buffer shall then be emptied before returning the TERMINAL RESPONSE to the UICC and allowing new UICC sending.

This mechanism to indicate single or concatenated data packets shall also be used by the UICC for launch parameters for a terminal application.

Upon receiving this command, the terminal shall either immediately send data or store provided data into the Tx buffer corresponding to the Channel identifier. Examples are given below, but the list is not exhaustive.

If the terminal is unable to process the command:

- if the command is rejected because the requested channel is already closed the terminal informs the UICC using TERMINAL RESPONSE (Bearer Independent Protocol error - channel identifier not valid);
- if the command is rejected because the terminal could not establish the link (after OPEN CHANNEL (on demand)) or the link was dropped, the terminal informs the UICC using TERMINAL RESPONSE (Bearer Independent Protocol error - channel closed);
- if the command is rejected because the channel is temporarily unavailable, the terminal informs the UICC using TERMINAL RESPONSE (terminal currently unable to process command);
- if the requested number of bytes of empty space is not yet available in the buffer the terminal informs the UICC using TERMINAL RESPONSE (Bearer Independent Protocol error);
- if the user has indicated the need to end the proactive UICC session, the terminal informs the UICC using TERMINAL RESPONSE (Proactive UICC session terminated by the user);
- if the command was the last or only command containing launch parameters and the command is rejected because the terminal prompted the user who rejected the start of the application, the terminal informs the UICC using TERMINAL RESPONSE (User did not accept the proactive command);
- if the command was the last or only command containing launch parameters and if the terminal has information about the launch parameters required by an application to be launched and the command is rejected because the terminal could not launch the application due to missing or unusable launch parameters, the terminal informs the UICC using TERMINAL RESPONSE (Bearer Independent protocol error, launch parameters missing or incorrect);
- if the command was the last or only command containing launch parameters and the command is rejected because the terminal could not launch the application due to any other reason, the terminal informs the UICC using TERMINAL RESPONSE (Bearer Independent protocol error, application launch failed).

If launching a terminal application with additional launch parameters failed, the UICC shall close the channel and may retry to launch the application afterwards.



If the terminal is able to process the command:

- if the requested number of bytes of empty space is available in the buffer the terminal shall inform the UICC that the command has been successfully executed, using **TERMINAL RESPONSE** and return the number of bytes of empty space available in the Tx buffer (or FF if more than 255 bytes are available);
- in the case of packet/datagram transmission, the structure of the SDU sent by the UICC to the terminal shall be fully respected while sending to the terminal external interface. The size of the SDU is therefore limited by the size of the packet PDU sent over the terminal external interface. In order to send one complete SDU, the CAT application may fill the Tx buffer with several **SEND DATA** commands, if necessary. Then the terminal shall send the complete SDU in one packet PDU;
- optionally, the UICC may include in the command an alpha identifier. The use of this alpha identifier by the terminal is described below and applies for the case where the command is not the last or only command containing launch parameters:
  - if the alpha identifier is provided by the UICC, the terminal shall use it to inform the user. The terminal may also use it to inform the user during data transfer. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
  - if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '0' and no value part), this is an indication that the terminal should not inform the user during data transfer;
  - if the alpha identifier is not provided by the UICC, the terminal may inform the user during data transfer.
- if an alpha identifier is included in the command and the command is the last or only command containing launch parameters, use of this alpha identifier by the terminal shall be as specified for the **OPEN CHANNEL** command related to Terminal Server Mode for the case of an application to be launched without additional launch parameters.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

A terminal of type NK or type ND may not alert the user and may start the application without explicit confirmation by the user.

### 6.4.31 GET CHANNEL STATUS

This clause applies if class "e" is supported.

This command requests the terminal to return a Channel status data object for each dedicated Channel identifier.

The terminal shall return the requested information concerning the channel(s) within a **TERMINAL RESPONSE** command.

### 6.4.32 SERVICE SEARCH

This clause applies if class "f" is supported.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

This command is used to search for the availability of a service in the environment of the terminal.

The UICC may provide a Device Filter. The devices responding to the service search shall then be part of the set given by Device Filter. If the Device Filter parameter is not present, no filter on the type of equipment is done by the terminal.

The UICC provides a Service Search parameter. The devices responding to the service search shall then support the requested service.

Upon receiving this command, the terminal shall decide if it is able to execute the command. Examples are given below, but the list is not exhaustive:

- if the command is rejected because the terminal is busy on a call, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - terminal currently busy on call);
- if the command is rejected because the bearer provided in the command is not available, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - bearer unavailable).

If the terminal is able to execute the command:

- the terminal performs the service search, gathers all received responses and informs the UICC using TERMINAL RESPONSE (command performed successfully, Service Availability);
- if the command fails because no device in the radio range supported the requested service, the terminal informs the UICC using TERMINAL RESPONSE (Bearer independent protocol error - Service error);
- if the command fails because there is no device reachable, the terminal informs the UICC using TERMINAL RESPONSE (Bearer independent protocol error - Remote device is not reachable).

### 6.4.33 GET SERVICE INFORMATION

This clause applies if class "f" is supported.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

This proactive command is used to look for the complete service record related to a service. By service record, it is meant all information that allows the UICC to define precisely the service (e.g. protocol stacks).

The UICC provides the Attribute Information parameter which indicates which detailed information is required.

Upon receiving this command, the terminal shall decide if it is able to execute the command. Examples are given below, but the list is not exhaustive.

If the command is rejected because the terminal is busy on a call, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - terminal currently busy on call).

If the command is rejected because the bearer provided in the command is not available, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - bearer unavailable).

If the terminal is able to execute the command:

- the terminal performs the search for the service details and informs the UICC using TERMINAL RESPONSE (command performed successfully, Service Record). The Service Record shall then be used as argument of an Open Channel proactive command.

If the command fails because there is no device reachable, the terminal informs the UICC using TERMINAL RESPONSE (Bearer independent protocol error - Remote device is not reachable).

If the CAT application already has all information concerning the service, it may directly try to connect the service performing an OPEN CHANNEL, and bypass the GET SERVICE INFORMATION step.

## 6.4.34 DECLARE SERVICE

This clause applies if class "f" is supported.

This command allows the UICC to download into the terminal service database the services that the card provides as a server. The declaration is to be made on a service by service basis, at the set up (e.g. after the profile download). The UICC shall indicate whether the terminal is required to add a new service in the terminal service database or to remove a service from the terminal service database.

When adding a new service, the UICC shall provide a Service Record that the terminal is required to register into its local service database.

When removing a service, the UICC shall provide the Service Identifier which uniquely identifies the service to be deleted from the database.

Upon receiving this command, the terminal shall decide if it is able to execute the command. Examples are given below, but the list is not exhaustive.

If the command is rejected because the terminal is busy on a call, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - terminal currently busy on call).

If the command is rejected because the terminal has not enough memory available to store the service record, the terminal informs the UICC using TERMINAL RESPONSE (Bearer Independent Protocol Error - Requested buffer size not available).

If the command for deletion is rejected because the service identifier is not valid, the terminal informs the UICC using TERMINAL RESPONSE (Bearer Independent Protocol Error - Service identifier unknown).

If the command is performed with modification of certain parameters of the Service Record (of which value is dynamically assigned by the terminal), the terminal informs the UICC using TERMINAL RESPONSE (command performed with modification).

If the terminal is able to execute the command: the terminal shall inform the UICC that the command has been successfully performed using TERMINAL RESPONSE (command performed successfully).

NOTE: A service can be coded using a coding type issued from a specific local bearer technology (e.g. Bluetooth or IrDA); however this service is to be considered by the terminal as available for any bearer.

## 6.4.35 SET FRAMES

This clause applies only if class "i" is supported.

For a terminal of type ND the support of this command is optional.

This command instructs the terminal to divide the terminal's screen into multiple, scrollable rectangular regions called frames in order to present multiple information at once.

This command offers the following possibilities:

- 1) Elements that the user should always see, such as copyright notices, and title graphics can be placed in a static, individual frame using the SETUP IDLE MODE TEXT command or DISPLAY TEXT in sustained mode.
- 2) Services are more functional. Several technologies may interact together (e.g. LAUNCH BROWSER becomes independent once launched). One frame may have the focus on the browser and another one on the SAT application. As the user navigates the site in "live" frames, the static frame's contents remain fixed.
- 3) Frames side-by-side design allows queries to be posed and answered on the same page, with one frame holding the query form, and the other presenting the results.

The SET FRAME command can be applied to the entire screen or to an already existing frame, dividing this frame into sub-frames.

If the Frame Identifier is the entire terminal screen, then, the terminal shall treat this as an instruction to clear the current frames (if applicable) and replace it with the new indicated frame layout.

If the Frame Identifier is the entire terminal screen and no frame layout is given, the terminal shall close all the frames and restore the main window on the terminal's screen. All information that is still associated with a frame (like idle mode text) is discarded.

The frames shall be removed from the terminal's display if either:

- the terminal is powered off; or
- the UICC is removed or a reset is performed.

The command has two arguments. These arguments specify the layout of the frames to be created: horizontal or vertical and their relative -size.

Vertical frames are created left-to-right, horizontal frames top-to-bottom. When a frame identifier is specified in the command, new frames are created in an existing frame but the total number of frames created shall be consistent with the "maximum number of frames supported" value given in the Terminal Profile.

Upon receiving this command, the terminal shall decide if it is able to execute the command. Examples are given below, but the list is not exhaustive:

- if the command is rejected because the terminal is not able to split its screen according to the parameters given, the terminal informs the UICC using TERMINAL RESPONSE (Frames error - requested size not supported);
- if the command is rejected because the number of frames required is higher than the number of frames supported by the terminal's screen, the terminal informs the UICC using TERMINAL RESPONSE (Frames error - number of frames beyond the terminal's capabilities);

If the terminal is able to set up the frames on the screen, the terminal shall:

- Divide the terminal's screen into multiple and scrollable regions defined by the parameters in the command.
- When frames are created, from a full screen or dividing an already existing frame, the content of the former frame (or full screen) shall be inserted in the new frame with the lowest frame identifier. The other frames shall be empty (blank screen).

NOTE: The content of each frame may then be updated using the usual commands.

## 6.4.36 GET FRAME STATUS

This clause applies if class "i" is supported.

For a terminal of type ND the support of this command is optional.

This command requests the terminal to return a Frames Parameters data object.

The terminal shall return the requested information concerning the frames within a TERMINAL RESPONSE command.

## 6.4.37 RETRIEVE MULTIMEDIA MESSAGE

This clause applies if class "j" is supported.

Upon receiving this command, the terminal shall decide if it is able to execute the command. Examples are given below, but the list is not exhaustive:

- if the command is rejected because the terminal is busy on a MMS transaction, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - currently busy on MMS transaction);
- if the command is rejected because the terminal is unable to process the MMS transaction, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - unable to process MMS transaction).

If the terminal is able to execute this command, the terminal shall:

- Retrieve the Multimedia Message from the network using the MMS message reference provided by the UICC in the Retrieve command parameters.
- Store the Multimedia Message on the UICC. The path of the file on the UICC in which the MM shall be stored is provided by the UICC in the Retrieve command parameters.
- Optionally, the UICC may include in this command an alpha-identifier. The use of this alpha-identifier by the terminal is described below:
  - if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it to inform the user. This is also an indication that the terminal should not give any other information to the user on the fact that the terminal is retrieving an MM. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
  - if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not give any information to the user on the fact that the terminal is retrieving an MM;
  - if the alpha identifier is not provided by the UICC, the terminal may give information to the user concerning what is happening.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

The storage completion shall be indicated in the ENVELOPE (MMS Transfer Status).

## 6.4.38 SUBMIT MULTIMEDIA MESSAGE

This clause applies if class "j" is supported.

Upon receiving this command, the terminal shall decide if it is able to execute the command. Examples are given below, but the list is not exhaustive:

- if the command is rejected because the terminal is busy on a MMS transaction, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - currently busy on MMS transaction);
- if the command is rejected because the terminal is unable to process the MMS transaction, the terminal informs the UICC using TERMINAL RESPONSE (terminal unable to process command - unable to process MMS transaction).

If the terminal is able to execute this command, the terminal shall:

- Get the Multimedia Message from the UICC. The path of the file on the UICC from which the MM shall be retrieved is provided by the UICC in the Submit command parameters.
- Submit the Multimedia Message to the network.
- Optionally, the UICC may include in this command an alpha-identifier. The use of this alpha-identifier by the terminal is described below:
  - if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it to inform the user. This is also an indication that the terminal should not give any other information to the user on the fact that the terminal is submitting an MM. If an icon is provided by the UICC, the icon indicated in the command may be used by the terminal to inform the user, in addition to, or instead of the alpha identifier, as indicated with the icon qualifier (see clause 6.5.4);
  - if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not give any information to the user on the fact that the terminal is submitting an MM;

- if the alpha identifier is not provided by the UICC, the terminal may give information to the user concerning what is happening.

A terminal of type ND shall ignore any alpha identifier provided together with this command. The terminal shall respond with "command performed successfully" upon successful completion of the command. A terminal of type ND shall also ignore any icon provided together with this command. The terminal shall respond with "command performed successfully but requested icon could not be displayed" upon successful completion of the command.

The submission status shall be indicated in the ENVELOPE (MMS Transfer Status).

## 6.4.39 DISPLAY MULTIMEDIA MESSAGE

This clause applies if class "j" is supported.

For a terminal of type ND the support of this command is optional.

This command shall be used to display a multimedia message. The multimedia message is defined in TS 123 140 [37].

This command allows the UICC to define the priority of the message. Two types of priority are defined:

- display normal priority multimedia message;
- display high priority multimedia message.

A flag (see command qualifier, clause 6.8.1) shall be set to inform the terminal whether the availability of the screen for subsequent information display after its use for "Display Multimedia Message" should be either after a short delay (the duration of the delay being at the discretion of the terminal manufacturer), or following a user MMI action.

An immediate response object may be included by the UICC, to indicate if the terminal should sustain the display beyond sending the TERMINAL RESPONSE.

The behaviour of Terminals supporting this feature is described below:

- if the user has indicated the need to end the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with "Proactive UICC session terminated by the user" result value;
- if the user has indicated the need to go backwards in the proactive UICC session, the terminal shall send a TERMINAL RESPONSE with "Backward move in the proactive UICC session requested by the user" result value;
- if a flag of the command qualifier (see clause 6.8.1) indicates that the terminal shall wait for the user to clear message and if the terminal decides that no user response has been received, the terminal shall send a TERMINAL RESPONSE with "No response from user" result value. A terminal of type NK which receives a DISPLAY MULTIMEDIA MESSAGE command with a command qualifier flag indicating "DISPLAY MULTIMEDIA MESSAGE - wait for user to clear" shall send a TERMINAL RESPONSE with "Command performed successfully" result value;
- if the UICC includes an immediate response object, the terminal shall immediately send TERMINAL RESPONSE (Command performed successfully). The terminal shall continue to display the multimedia message until one of the following events occurs:
  - a subsequent proactive command is received containing display data;
  - the expiration of the short delay, if so indicated by the command qualifier;
  - following a user MMI action;
  - when a higher priority event occurs, e.g. an incoming mobile terminated call.
- no further TERMINAL RESPONSE shall be sent when the terminal removes the multimedia message from the display, regardless of the cause;
- otherwise, the terminal shall send TERMINAL RESPONSE (Command performed successfully) at the apparition of either the short delay or the variable display timeout, or following a user MMI action not described above.

In each case the availability of the screen for the subsequent information display is defined in clause 6.9.

NOTE: For the case where the message is cleared after a short delay, the terminal may also allow the user to clear the display via the MMI prior to this.

The terminal shall reject normal priority text or multimedia messages commands if the screen is currently being used for more than its normal stand-by display. If the command is rejected, the terminal informs the UICC using TERMINAL RESPONSE (terminal currently unable to process command - screen busy).

High priority text or multimedia message should be displayed on the screen immediately, except if there is a conflict of priority level of alerting (e.g. emergency call, incoming calls, low battery warning). In that situation, the resolution is left to the terminal. If the command is rejected in spite of the high priority, the terminal shall inform the UICC using TERMINAL RESPONSE (terminal currently unable to process command - screen is busy).

If help information is requested by the user, this command may be used to display help information on the screen. The help information should be sent as high priority message and with the option that it should be cleared after a short delay.

#### 6.4.40 ACTIVATE

This clause applies if class "l" is supported.

This command requests the terminal to activate a specified interface.

If the terminal is able to activate the requested interface, or if the interface is already active, the terminal shall send back a TERMINAL RESPONSE (Command performed successfully).

#### 6.4.41 CONTACTLESS STATE CHANGED

This clause applies if class "r" is supported.

This command allows the UICC to inform the terminal when the contactless functionality in the UICC has been enabled or disabled. The contactless functionality may be enabled or disabled:

- when requested by the user via a CAT command; or
- in response to an event from the terminal as specified in clause 7.5.19.

A terminal supporting this proactive command shall provide an information element (icon, etc.) that indicates the state of the contactless functionality to the user. It shall update this indicator according to the information given in the command. After a card reset, the indicator in the terminal shall be set to disabled.

#### 6.4.42 COMMAND CONTAINER

This clause applies if class "u" is supported.

This command allows the UICC to send a CAT command to an eCAT client by encapsulation it in the command COMMAND CONTAINER. Any CAT command except COMMAND CONTAINER and ENCAPSULATED SESSION CONTROL may be encapsulated. The first pair of command details and device identities refer to the COMMAND CONTAINER and indicate as destination the eCAT client. The second pair of command details and device identities refer to the encapsulated CAT command.

When the terminal is able to process the COMMAND CONTAINER, it shall unwrap the encapsulated command and pass it to the indicated eCAT client. The eCAT client shall process the encapsulated command and return the terminal response to it. The terminal shall embed this response in its terminal response to the COMMAND CONTAINER.

#### 6.4.43 ENCAPSULATED SESSION CONTROL

This clause applies if class "u" is supported.

This command allows the UICC to end an encapsulated command session with an eCAT client.

If class "x" is supported in addition, this command allows the UICC to set an eCAT secure channel with an eCAT client or to terminate an eCAT secure channel with an eCAT client.

To terminate an eCAT secure channel, the ENCAPSULATED SESSION CONTROL command shall be sent in a secured COMMAND CONTAINER and the command details of the encapsulated command shall indicate "close Master and Connection SA, keep encapsulated command session".

## 6.5 Common elements in proactive UICC commands

### 6.5.1 Command number

The command number is to cater for the future possibility of multiple ongoing commands (i.e. when the UICC issues further commands before receiving the response to the ongoing command). The implications of such multiple ongoing commands have not been elaborated at this stage of the toolkit specification.

Each command issued by a proactive UICC during a card session shall have its own command number. Command numbers may take any hexadecimal value between "01" and "FE". The command number is held in the command details data object.

The UICC is responsible for assigning the command number.

The terminal shall keep a record of the status of each command and its command number, until the terminal gives the result of the command to the UICC, using TERMINAL RESPONSE. After this, the terminal may erase all internal records concerning this command. The command number is then free for allocation by the UICC to a new command.

When the terminal is powered off and on, the details of any ongoing command shall be reset. The terminal shall not be expected to know the status of commands issued in a previous card session.

### 6.5.2 Device identities

This data object gives the devices which are the source and destination for the instruction. Only certain combinations of source and destination devices are allowed for each proactive command. These are given in clause 10.

### 6.5.3 Alpha identifier

Many of the commands include an alpha identifier data object. The text it contains shall be displayed on screen by the terminal at the same time as the UICC command is performed.

### 6.5.4 Icon identifiers

Some commands may provide an icon identifier. Icons are intended to enhance the MMI by providing graphical information to the user. The display of icons is optional for the terminal. If icons are provided by the UICC, the related alpha identifier or text string shall be present and not a null string.

The UICC indicates to the terminal whether the icon replaces an alpha identifier or text string, or whether it accompanies it (see clause 8.32).

If both an alpha identifier or text string, and an icon are provided with a proactive command, and both are requested to be displayed, but the terminal is not able to display both together on the screen, then the alpha identifier or text string takes precedence over the icon.

If the UICC provides an icon identifier with a proactive command, then the terminal shall inform the UICC if the icon could not be displayed by sending the general result "Command performed successfully, but requested icon could not be displayed".

If the terminal receives an icon, and either an empty or no alpha identifier/text string is given by the UICC, then the terminal shall reject the command with general result "Command data not understood by terminal".

**NOTE:** Application designers should be aware that icons provided by the application may not be displayed by the terminal.



## 6.5.5 Text Attribute

Some commands may provide a text attribute. Text attributes are intended to enhance the MMI when providing information to the user. The display of various text formats as described in TS 123 040 [27] are optional for the terminal.

## 6.5.6 Frame identifier

Some commands may provide a frame identifier. Frames are intended to enhance the MMI by dividing the terminal's screen into several rectangular regions (frames). The display of frames is optional for the terminal.

If the UICC provides a frame identifier with a proactive command, then the terminal supporting the frames feature shall display the command proactive information (e.g. text string, alpha identifier, icon, etc.) in the corresponding frame. If the user terminates a proactive session, this shall only affect the frame in which the proactive command is executed.

If the screen is split into frames and no frame identifier is given in a proactive command or the frame identifier is invalid, the default frame shall be used.

If the terminal does not support the frames feature or the screen is not split into frames, it shall ignore the frame identifier data object.

## 6.6 Structure of proactive UICC commands

The general structure of proactive UICC commands using TLV objects is described in annex C.

### 6.6.1 DISPLAY TEXT

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Text string	8.15	M	Y	C
Icon identifier	8.31	O	N	D
Immediate response	8.43	O	N	E
Duration	8.8	O	N	F
Text Attribute	8.72	O	N	G
Frame Identifier	8.80	O	N	H

- Duration:
  - Contents:
    - the required duration for execution of the command before the timeout expires. Resolution and the precision of the time value are in accordance with clause 6.4.21.
- Text Attribute applies to the Text String.

## 6.6.2 GET INKEY

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Text string	8.15	M	Y	C
Icon identifier	8.31	O	N	D
Duration	8.8	O	N	E
Text Attribute	8.72	O	N	F
Frame Identifier	8.80	O	N	G

- Text string:
  - Contents:
    - text for the Terminal to display in conjunction with asking the user to respond.
- Duration:
  - Contents:
    - the duration for execution of the command before the timeout expires. Resolution and the precision of the time value are in accordance with clause 6.4.21.
- Text Attribute applies to the Text String.

## 6.6.3 GET INPUT

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Text string	8.15	M	Y	C
Response length	8.11	M	Y	D
Default Text	8.23	O	N	E
Icon identifier	8.31	O	N	F
Text Attribute	8.72	O	N	G
Frame Identifier	8.80	O	N	H

- Text string:
  - Contents:
    - text for the Terminal to display in conjunction with asking the user to respond.
  - Text Attribute applies to Text string when supported.
- Response length:
  - Contents:
    - the minimum and maximum acceptable lengths length in characters (see clause 6.4.3) for the response from the user.

- Default Text:
  - Contents:
    - text for the Terminal to display corresponds to a default text string offered by the UICC.
  - Text Attribute does not apply to Default text.

#### 6.6.4 MORE TIME

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B

#### 6.6.5 PLAY TONE

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Tone	8.16	O	N	D
Duration	8.8	O	N	E
Icon identifier	8.31	O	N	F
Text Attribute	8.72	C	N	G
Frame Identifier	8.80	O	N	H

- Tone:
  - Contents:
    - the standard supervisory tone or proprietary Terminal tone that the Terminal shall generate, either on its own or on top of the downlink audio path. If no tone is specified, then the Terminal shall default to "general beep".
- Duration:
  - Contents:
    - the length of time for which the Terminal shall generate the tone, if the tone is continuous or repeatable. For single tones, the value of this data object shall be ignored by the Terminal. If no duration is specified, the Terminal shall default to a duration determined by the Terminal manufacturer.
- Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier data object is present.

#### 6.6.6 POLL INTERVAL

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Duration	8.8	M	Y	C

- Duration:
  - Contents:
    - the maximum interval between two STATUS commands related to Proactive Polling.

## 6.6.7 SET-UP MENU

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D1+D2+... Dn+E+F+G+H+I)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	M	Y	C
Item data object for item 1	8.9	M	Y	D1
Item data object for item 2	8.9	O	N	D2
...	8.9	O	N	Dx
Item data object for last item in list	8.9	O	N	Dn
Items Next Action Indicator	8.24	O	N	E
Icon identifier	8.31	O	N	F
Item Icon identifier list	8.32	O	N	G
Text Attribute	8.72	O	N	H
Item Text Attribute List	8.73	O	N	I

The SET-UP MENU command BER-TLV data object shall contain Item COMPREHENSION-TLV data objects. Each Item data object contains an item in the list, for the user to choose. The length of each Item data object may be different. Within a list, each Item shall have a unique item identifier.

If the "Item data object for item 1" is a null data object (i.e. length = '00' and no value part), this is an indication to the terminal to remove the existing menu from the menu system in the terminal.

If the UICC provides an Items Next Action Indicator data object, the comprehension required flag shall be set to '0'.

The UICC may provide a title icon identifier data object and/or an item icon identifier list data object. The item icon identifier data object contains an icon identifier for each item.

The UICC provides a title (Alpha Identifier) with a Text Attribute data object and/or an item Text Attribute list data object. The item Text Attribute list data object contains a Text Attribute for each item.

## 6.6.8 SELECT ITEM

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D1+D2+... Dn+E+F+G+H+I+J+K)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Item data object for item 1	8.9	M	Y	D1
Item data object for item 2	8.9	O	N	D2
...	8.9	O	N	Dx
Item data object for last item in list	8.9	O	N	Dn
Items Next Action Indicator	8.24	O	N	E
Item Identifier	8.10	O	N	F
Icon identifier	8.31	O	N	G
Item Icon identifier list	8.32	O	N	H
Text Attribute	8.72	C	N	I
Item Text Attribute List	8.73	O	N	J
Frame Identifier	8.80	O	N	K

The SELECT ITEM command BER-TLV data object shall contain Item COMPREHENSION-TLV data objects. Each Item data object contains an item in the list, for the user to choose. The length of each Item data object may be different.

Within a list, each Item shall have a unique item identifier. The SELECT ITEM command BER-TLV data object may contain a single Item Identifier data object as an indication of the default item. The Comprehension Required flag in the Item Identifier data object shall be set to 0, indicating that it is not mandatory for the terminal to support indication of the default item.

If the UICC provides an Items Next Action Indicator data object, the comprehension required flag shall be set to "0".

The UICC may provide a title icon identifier data object and/or an item icon identifier list data object. The item icon identifier data object contains an icon identifier for each item.

The Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present. The item Text Attribute list data object contains a Text Attribute for each item.

### 6.6.9 SEND SHORT MESSAGE

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H+I)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Address	8.1	C	N	D
3GPP-SMS TPDU	8.13	C	Y/N	E
CDMA-SMS TPDU	8.71	C	Y/N	F
Icon identifier	8.31	O	N	G
Text Attribute	8.72	C	N	H
Frame Identifier	8.80	O	N	I

The address data object may hold the address of the Service Centre for some access technologies. If no address is transferred and an address is needed, then the terminal shall insert the default Service Centre address.

Either 3GPP-SMS TPDU or CDMA-SMS TPDU shall be present in the SEND SHORT MESSAGE command. The UICC shall not send the 3GPP-SMS TPDU and the CDMA-SMS TPDU in the same command.

The Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

### 6.6.10 Void

### 6.6.11 Void

### 6.6.12 SET UP CALL

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H+I+J+K+L+M)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier (user confirmation phase)	8.2	O	N	C
Address	8.1	M	Y	D
Capability configuration parameters	8.4	O	N	E
Subaddress	8.3	O	N	F
Duration	8.8	O	N	G
Icon identifier (user confirmation phase)	8.31	O	N	H
Alpha identifier (call set up phase)	8.2	O	N	I
Icon identifier (call set up phase)	8.31	O	N	J
Text Attribute (user confirmation phase)	8.72	C	N	K
Text Attribute (call set up phase)	8.72	C	N	L
Frame Identifier	8.80	O	N	M

If the capability configuration parameters are not present, the terminal shall assume the call is a speech call.

If the subaddress is not present, the terminal shall not provide a called party subaddress to the network.

If the duration is not present, the UICC imposes no restrictions on the terminal of the maximum duration of redials.

The Text Attribute (user confirmation phase) applies to the Alpha Identifier (user confirmation phase). The Text Attribute (call set up phase) applies to the Alpha identifier (call set up call phase). One Text Attribute may be present only if at least one Alpha Identifier is present. Both Text Attributes may be present only if both Alpha Identifiers are present. If only one Text Attribute data object is present, it shall apply to the first or unique Alpha identifier present in the command.

### 6.6.13 REFRESH

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
File List	8.18	C	N	C
AID	8.60	O	N	D
Alpha identifier	8.2	O	N	E
Icon identifier	8.31	O	N	F
Text Attribute	8.72	C	N	G
Frame Identifier	8.80	O	N	H

For the refresh modes "File Change Notification", "NAA Initialization and File Change Notification" and "NAA Session Reset", the UICC shall supply a File List data object, indicating which eFs need to be refreshed. For other modes, inclusion of a File List is optional, and the terminal shall ignore it.

If an AID TLV is present, it indicates the NAA which needs to be refreshed. If it is not present, the terminal shall assume the current NAA needs to be refreshed. The AID COMPREHENSION-TLV can only be present on a 3G platform, it shall not be provided on a 2G platform.

The Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

### 6.6.14 POLLING OFF

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B

### 6.6.15 PROVIDE LOCAL INFORMATION

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B

### 6.6.16 SET UP EVENT LIST

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
Event list	8.25	M	Y	C

If the Event list is a null data object (i.e. length = '00' and no value part), this is an indication to the terminal to remove the existing list of events in the terminal.

### 6.6.17 PERFORM CARD APDU

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
C-APDU	8.35	M	Y	C

### 6.6.18 POWER OFF CARD

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B

### 6.6.19 POWER ON CARD

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B

### 6.6.20 GET READER STATUS

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B

## 6.6.21 TIMER MANAGEMENT

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
Timer Identifier	8.37	M	Y	C
Timer value	8.38	C	N	D

- Timer Identifier:
  - Contents:
    - identifier of the timer to which the command applies.
- Timer value:
  - Contents:
    - length of time during which the timer has to run. The UICC shall supply this data object only when a timer has to be started.

## 6.6.22 SET UP IDLE MODE TEXT

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	8.2	M	Y	1
Length (A+B+C+D+E+F)	-	M	Y	1 or 2
Command details	7.5.6	M	Y	A
Device identities	7.5.7	M	Y	B
Text string	7.5.15	M	Y	C
Icon identifier	8.31	O	N	D
Text Attribute	8.72	O	N	E
Frame Identifier	8.80	O	N	F

If the "Text string" is a null data object (i.e. length = '00' and no value part), the terminal shall remove the existing idle mode text in the terminal.

The Text Attribute applies to the Text String.

## 6.6.23 RUN AT COMMAND

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
Alpha Identifier	8.2	O	N	C
AT Command	8.40	M	Y	D
Icon identifier	8.31	O	N	E
Text Attribute	8.72	C	N	F
Frame Identifier	8.80	O	N	G

The Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.



## 6.6.24 SEND DTMF COMMAND

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
Alpha Identifier	8.2	O	N	C
DTMF String	8.44	M	Y	D
Icon identifier	8.31	O	N	E
Text Attribute	8.72	C	N	F
Frame Identifier	8.80	O	N	G

The Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

## 6.6.25 LANGUAGE NOTIFICATION

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C)	-	M	Y	1
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Language	8.45	C	Y/N	C

- Language:
  - Contents:
    - currently used language. The UICC shall include a Language object, when a specific language is being notified.

## 6.6.26 LAUNCH BROWSER

Description	Clause	M/O	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F1+F2+...+FN+G+H+I+J+K+L+M+N)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
Browser Identity	8.47	O	N	C
URL	8.48	M	Y	D
Bearer	8.49	O	N	E
Provisioning File Reference 1	8.50	O	N	F1
Provisioning File Reference 2	8.50	O	N	F2
...	8.50	O	N	Fx
Provisioning File Reference N	8.50	O	N	FN
Text String (Gateway/Proxy Identity)	8.15	O	N	G
Alpha identifier (user confirmation phase)	8.2	O	N	H
Icon identifier (user confirmation phase)	8.31	O	N	I
Text Attribute	8.72	C	N	J
Frame Identifier	8.80	O	N	K
Network Access Name	8.70	O	N	L
Text String (User login)	8.15	O	N	M
Text String (User password)	8.15	O	N	N

If the URL data object is provisioned the URL value shall take precedence over any other URL value.

If Provisioning File Reference data object is present in the command then it shall take precedence over Bearer, Proxy Identity and Network Access Name with related "User login" and "User password". If several Provisioning File References are present in the same command the information in the first reference shall take precedence.

Gateway/Proxy Identity is a text string which gives to the terminal the name/identity of the Gateway/Proxy to be used for connecting to the URL. This Gateway/Proxy Identity is required when the bearer data object is present.

Text Attribute applies to the alpha identifier (user confirmation phase). It may be present only if the Alpha Identifier (user confirmation phase) is present.

If the Network Access Name is present, the terminal shall use the Network Access Name value to identify the Gateway entity to be used during this launched browsing session. The value shall be valid only for the session initiated by this Launch Browser command and shall not be used for subsequent sessions. If Network Access Name is not present, the terminal shall use the default Network Access Name in the terminal configuration or the default subscription value.

If the Network Access Name is present in the command the UICC may provide "User login" and "User password" parameters, which can be used for authentication. If the Network Access Name is present but such parameters are not specified, the access will be made without authentication.

If the Network Access Name is not present, "User login" and "User Password" shall not be present.

## 6.6.27 OPEN CHANNEL

### 6.6.27.1 OPEN CHANNEL related to CS bearer

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H+I+J+K+L+M+N+O+P+Q)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
Address	8.1	M	Y	E
Subaddress	8.3	O	N	F
Duration 1	8.8	C	N	G
Duration 2	8.8	O	N	H
Bearer description	8.52	M	Y	I
Buffer size	8.55	M	Y	J
Other address (local address)	8.58	O	N	K
Text String (User login)	8.15	O	N	L
Text String (User password)	8.15	O	N	M
UICC/terminal interface transport level	8.59	O	N	N
Data destination address	8.58	C	Y	O
Text Attribute	8.72	C	N	P
Frame Identifier	8.80	O	N	Q

The subaddress may be requested. If the subaddress is not present, the terminal shall not provide a called party subaddress to the network.

Duration 1 indicates the duration of reconnection tries. If Duration 1 is not present or is null, the UICC imposes no restrictions on the terminal. Duration 1 shall be present if Duration 2 is present.

Duration 2 indicates the timeout value before the terminal releases the link if there is no data exchanged on the link. If duration 2 is not present the link is never released automatically by the terminal.

The local address parameter provides information to the terminal necessary to identify the local device (i.e. it provides an IP address). If local address length is null, dynamic local address is required. If parameter is not present, the terminal may use the terminal default local address configuration.

The terminal may support a remote access login feature (e.g. PPP login). If supported by the terminal, the UICC may provide "User login" and "User password" parameters which allow the terminal to answer an access authentication challenge. If only one parameter is present, it is considered as the User Login and the terminal shall use default Password configuration if any. If the parameters are not present, the terminal shall use default Login/Password configuration if any. If no authentication challenge is requested, the user login and password parameters shall be ignored.

If the UICC/terminal interface transport level is present in the command, then the terminal shall provide the requested transport layer protocols under the channel and shall use this object containing a set of parameters required to make the transport connection. The data that is exchanged at the UICC/terminal interface in the RECEIVE DATA/SEND DATA commands are SDUs. When the CAT application sends an SDU, the transport layer within the terminal is in charge to add the transport header to the SDU in order to build the Transport-PDU. When the CAT application requests to receive an SDU, the transport layer within the terminal is in charge to remove the transport header of the Transport-PDU, and to forward the SDU to the CAT. If the parameter is not present, the UICC/terminal interface is the bearer level (serial link or packet link), and the CAT application is in charge of the network and transport layer. If present, the UICC/terminal interface transport level shall either be set to UDP, UICC in client mode, remote connection or to TCP, UICC in client mode, remote connection.

The Data destination address is the end point destination address of sent data. This data destination address is requested when a UICC/terminal interface transport is present, otherwise it is ignored. The data destination address is a data network address.

Text Attribute applies to the alpha identifier. It may be present only if the Alpha Identifier is present.

### 6.6.27.2 OPEN CHANNEL related to packet data service bearer

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H+I+J+K+L+M+N)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
Bearer description	8.52	M	Y	E
Buffer size	8.55	M	Y	F
Network Access Name	8.70	O	N	G
Other address (local address)	8.58	O	N	H
Text String (User login)	8.15	O	N	I
Text String (User password)	8.15	O	N	J
UICC/terminal interface transport level	8.59	O	N	K
Data destination address	8.58	C	Y	L
Text Attribute	8.72	C	N	M
Frame Identifier	8.80	O	N	N

The Network Access Name may be requested. The Network Access Name provides information to the terminal necessary to identify the Gateway entity which provides interworking with an external packet data network. If the parameter is not present, the terminal may use the default Network Access Name in the terminal configuration or the default subscription value.

The local address parameter provides information to the terminal necessary to identify the local device. If the parameter is present and length is not null, it provides an IP address that identifies the CAT application in the address area applicable to the PDN. If local address length is null, dynamic local address allocation is required for the CAT application. If parameter is not present, the terminal may use the terminal default local address configuration.

The TE may support a remote access login feature. If supported by the TE, the UICC may provide "User login" and "User password" parameters, which can be used for authentication. If only one parameter is present, it is considered as the User Login and the TE shall use default Password configuration if any. If the parameters are not present, the TE shall use default Login/Password configuration if any. If no authentication is requested, the user login and password parameters shall be ignored.

If the UICC/terminal interface transport level is present in the command, then the terminal shall provide the requested transport layer protocols under the channel and shall use this object containing a set of parameters required to make the transport connection. The data that is exchanged at the UICC/terminal interface in the RECEIVE DATA/SEND DATA commands are SDUs. When the CAT application sends an SDU, the transport layer within the terminal is in charge to add the transport header to the SDU in order to build the Transport-PDU. When the SAT application requests to receive an SDU, the transport layer within the terminal is in charge to remove the transport header of the Transport-PDU, and to forward the SDU to the CAT. If the parameter is not present, the UICC/terminal interface is the bearer level (serial link or packet link), and the CAT application is in charge of the network and transport layer. If present, the UICC/terminal interface transport level shall either be set to UDP, UICC in client mode, remote connection or to TCP, UICC in client mode, remote connection.

The Data destination address is the end point destination address of sent data. This data destination address is requested when a UICC/terminal interface transport is present, otherwise it is ignored. The data destination address is a data network address (e.g. IP address).

Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

### 6.6.27.3 OPEN CHANNEL related to local bearer

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H+I+J+K+L+M+N)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
Duration 1	8.8	C	N	E
Duration 2	8.8	O	N	F
Bearer description	8.52	M	Y	G
Buffer size	8.55	M	N	H
Text String (User password)	8.15	O	N	I
UICC/terminal interface transport level	8.59	O	N	J
Data destination address	8.58	C	Y	K
Remote Entity Address	8.68	O	N	L
Text Attribute	8.72	C	N	M
Frame Identifier	8.80	O	N	N

Duration 1 indicates the duration of reconnection tries. If Duration 1 is not present or is null, the UICC imposes no restrictions on the terminal. Duration 1 shall be present if Duration 2 is present.

Duration 2 indicates the timeout value before the terminal releases the link if there is no data exchanged on the link. If duration 2 is not present the link is never released automatically by the terminal.

Bearer Description gives detailed information characterizing the bearer. When the UICC acts as a server, local information (local service record data) is included in Bearer Description; in addition, if the UICC provides a Service Record field (which is part of the Bearer Description TLV) different from '00', the terminal shall ignore it and proceed with the command. When the UICC acts as a client, remote information (remote service record data) is included in Bearer Description; in addition, if the UICC provides a Service Identifier field (which is part of the Bearer Description TLV) different from 'FF', the terminal shall ignore it and proceed with the command.

The UICC may optionally provide a user password that should be used by the terminal for authentication. For the Bluetooth local bearer, the user password corresponds to the passkey/PIN as defined in the Bluetooth specification [16].

If the UICC/terminal interface transport level is present in the command, then the terminal shall provide the requested transport layer protocols under the channel and shall use this object containing a set of parameters required to make the transport connection. If the parameter is not present, the UICC/terminal interface is the bearer level. The data that will be received/sent from the SAT to the transport layer is a SDU that will be received/transmitted in the Transport-PDU. If present, the UICC/terminal interface transport level shall either be set to UDP, UICC in client mode, remote connection or to TCP, UICC in client mode, remote connection.

The Data destination address is the end point destination address of sent data. This data destination address is requested when a UICC/terminal interface transport is present, otherwise it is ignored. The data destination address is a data network address (e.g. IP address).

The Remote Entity Address parameter provides information to the terminal necessary to identify the entity which provides access to the requested resource. Depending on the local technology, this parameter is necessary or not. For Bluetooth, it shall be the BD\_ADDR of the remote device. For IrDA, it shall be the 32 bits address of the remote device.

Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

#### 6.6.27.4 OPEN CHANNEL related to Default (network) Bearer

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H+I+J+K+L+M)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
Bearer description	8.52	M	Y	E
Buffer size	8.55	M	Y	F
Other address (local address)	8.58	O	N	G
Text String (User login)	8.15	O	N	H
Text String (User password)	8.15	O	N	I
UICC/terminal interface transport level	8.59	O	N	J
Data destination address	8.58	C	Y	K
Text Attribute	8.72	C	N	L
Frame Identifier	8.80	O	N	M

The local address parameter (see clause 8.58) provides information to the terminal necessary to identify the local device. If the parameter is present and length is not null, it provides an IP address that identifies the USAT application in the address area applicable to the PDN. If local address length is null, dynamic local address allocation is required for the USAT application. If parameter is not present, the mobile may use the mobile default local address configuration.

The terminal may support a remote access login feature. If supported by the terminal, the UICC may provide "User login" and "User password" parameters, which can be used for authentication. If only one parameter is present, it is considered as the User Login and the terminal shall use default Password configuration if any. If the parameters are not present, the terminal shall use default Login/Password configuration if any. If no authentication challenge is requested, the user login and password parameters shall be ignored.

If the UICC/terminal interface transport level is present in the command, then the terminal shall provide the requested transport layer protocols under the channel and shall use this object containing a set of parameters required to make the transport connection. The data that is exchanged at the UICC/terminal interface in the RECEIVE DATA/SEND DATA commands are SDUs. When the USAT application sends an SDU, the transport layer within the terminal is in charge to add the transport header to the SDU in order to build the Transport-PDU. When the USAT application requests to receive an SDU, the transport layer within the terminal is in charge to remove the transport header of the Transport-PDU, and to forward the SDU to the USAT. If the parameter is not present, the UICC/terminal interface is the bearer level (serial link or packet link as defined in TS 127 007 [5]) and the USAT application is in charge of the network and transport layer. If present, the UICC/terminal interface transport level shall either be set to UDP, UICC in client mode, remote connection or to TCP, UICC in client mode, remote connection.

The Data Destination Address is the end point destination address of sent data. This Data Destination Address is requested when a UICC/terminal interface transport level is present, otherwise it is ignored. The Data Destination Address is a data network address (e.g. IP address).

Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

## 6.6.27.5 OPEN CHANNEL related to UICC Server Mode

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
Buffer size	8.55	M	Y	E
UICC/terminal interface transport level	8.59	M	Y	F
Text attribute	8.72	C	N	G
Frame identifier	8.80	O	N	H

The UICC/terminal interface transport level shall be set to "TCP, UICC in server mode".

Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

## 6.6.27.6 OPEN CHANNEL related to Terminal Server Mode

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Buffer size	8.55	M	Y	C
UICC/terminal interface transport level	8.59	M	Y	D
Alpha identifier	8.2	O	N	E
Icon identifier	8.31	O	N	F
Text Attribute	8.72	C	N	G
Frame Identifier	8.80	O	N	H

The UICC/terminal interface transport level shall be set to "TCP, UICC in client mode, local connection" or "UDP, UICC in client mode, local connection" or "direct communication channel". The UICC shall only use "direct communication channel" in the command if the terminal indicated in the terminal profile that it supports the direct communication channel for OPEN CHANNEL in Terminal Server Mode.

The UICC shall only use the setting "launch parameters following" in the command details if the terminal indicated in the terminal profile that it supports launch parameters for OPEN CHANNEL in Terminal Server Mode.

The UICC shall only include the alpha identifier and/or icon identifier and/or text attribute and/or frame identifier data object(s) in the command if the terminal indicated in the terminal profile that it supports confirmation parameters for OPEN CHANNEL in Terminal Server Mode.

Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

## 6.6.28 CLOSE CHANNEL

Description	Clause	M/O	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
Text Attribute	8.72	C	N	E
Frame Identifier	8.80	O	N	F

The Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

## 6.6.29 RECEIVE DATA

Description	Clause	M/O	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
Channel data length	8.54	M	Y	E
Text Attribute	8.72	C	N	F
Frame Identifier	8.80	O	N	G

The Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

## 6.6.30 SEND DATA

Description	Clause	M/O	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
Channel data	8.53	M	Y	E
Text Attribute	8.72	C	N	F
Frame Identifier	8.80	O	N	G

The Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

## 6.6.31 GET CHANNEL STATUS

Description	Clause	M/O	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B

## 6.6.32 SERVICE SEARCH

Description	Clause	M/O	Min	Length
Proactive UICC command Tag	9.3	M	Y	1
Length (A+B+C+D+E+F+G+H)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
Service search	8.65	M	Y	E
Device filter	8.64	O	N	F
Text Attribute	8.72	C	N	G
Frame Identifier	8.80	O	N	H

The Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

### 6.6.33 GET SERVICE INFORMATION

Description	Clause	M/O	Min	Length
Proactive UICC command Tag	9.3	M	Y	1
Length (A+B+C+D+E+F+G)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
Attribute information	8.66	M	Y	E
Text Attribute	8.72	C	N	F
Frame Identifier	8.80	O	N	G

The Text Attribute applies to the Alpha Identifier. It may be present only if the Alpha Identifier is present.

### 6.6.34 DECLARE SERVICE

Description	Clause	M/O	Min	Length
Proactive UICC command Tag	9.3	M	Y	1
Length (A+B+C+D)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
Service Record	8.63	M	Y	C
UICC/terminal interface	8.59	O	N	D

For Device identities field, Destination Device Identity is required to be the terminal.

The UICC/terminal interface parameter specifies the protocol stack the UICC will be connected to on the terminal.

If the UICC/terminal interface data object is not present, the UICC/terminal interface is the bearer level as defined in the OPEN CHANNEL command.

### 6.6.35 SET FRAMES

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Frame Identifier	8.80	M	Y	C
Frame Layout	8.78	C	Y	D
Default Frame Identifier	8.80	O	N	E

The frame identifier '00' refers to the entire terminal's screen.

If the frame identifier is '00' and the frame layout is not present, all frames shall be removed from the screen.

If the frame identifier is '00' and the frame layout is present, any existing frame shall be removed from the screen, the default frame information shall be reset or set to the new value, and the entire screen shall be split according to the new frame layout.

A frame identifier different from '00' refers to an existing frame. In this case, the frame layout is mandatory and defines how this frame shall be split up into (sub-)frames.



Frame identifiers shall be allocated as follows:

- If the entire screen is split up into N frames, these frames shall be numbered from 1 to N starting from top or left.
- If K frames are already existing and frame M is split into N (sub-)frames, frames 1 to M-1 shall keep their number, the numbers of frames M+1 to K shall be increased by N-1, and the newly created frames shall be assigned the numbers M to M+N-1, starting from top or left.

The SET FRAMES command BER-TLV data object may contain a Default Frame data object as an indication of the frame to be used to display information in case where the Frame Identifier is not included in the proactive commands.

If Default Frame was not present in any SET FRAMES command since the last split up of the entire screen, the terminal shall use the frame with identifier "01" as default frame. If several SET FRAME commands since the last split up of the entire screen included a default frame object, the last one is valid. If frames are re-numbered as defined above, this also applies to the default frame.

### 6.6.36 GET FRAMES STATUS

Description	Clause	M/O	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B

### 6.6.37 RETRIEVE MULTIMEDIA MESSAGE

Description	Clause	M/O	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H+I+J)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
Multimedia Message Reference	8.82	M	Y	E
MMS Reception File	8.18	M	Y	F
MM Content Identifier	8.85	M	Y	G
Multimedia Message Identifier	8.83	C	N	H
Text Attribute	8.72	C	N	I
Frame Identifier	8.80	O	N	J

Multimedia Message Reference is the "MM1\_retrieve.REQ" (see TS 123 140 [37]) message that is needed for the retrieval of the multimedia message and it contains the URI identifying the multimedia message in the network.

MMS Reception File is a path of a file on the UICC. This path shall be used by the terminal once the MM is retrieved from the network to store the MM on the UICC.

Multimedia Message Identifier is the identifier of the Multimedia Message within the MMS Reception File.

Text Attribute applies to the alpha identifier. It may be present only if the Alpha Identifier is present.

A terminal response shall be sent immediately upon reception of the command and shall not wait for any response from the network.

### 6.6.38 SUBMIT MULTIMEDIA MESSAGE

Description	Clause	M/O	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F+G+H)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Alpha identifier	8.2	O	N	C
Icon identifier	8.31	O	N	D
MMS Submission File	8.18	M	Y	E
Multimedia Message Identifier	8.83	C	N	F
Text Attribute	8.72	C	N	G
Frame Identifier	8.80	O	N	H

MMS Submission File is a path of a file on the UICC. This path shall be used by the terminal to get the MM from the UICC and then to submit it to the network.

Multimedia Message Identifier is the identifier of the Multimedia Message within the MMS Submission File. This Identifier is mandatory in case the MMS Submission File is able to store several MMs.

Text Attribute applies to the alpha identifier. It may be present only if the Alpha Identifier is present.

A terminal response shall be sent immediately upon reception of the command and shall not wait for any response from the network.

### 6.6.39 DISPLAY MULTIMEDIA MESSAGE

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
MMS Submission File	8.18	M	Y	C
Multimedia Message identifier	8.83	M	Y	D
Immediate response	8.43	O	N	E
Frame Identifier	8.80	O	N	F

### 6.6.40 ACTIVATE

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Activate descriptor	8.89	M	Y	C

### 6.6.41 CONTACTLESS STATE CHANGED

This proactive command allows the UICC to inform the terminal of a change in the state of the contactless functionality in the UICC.

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C)	-	M	Y	1
Command details	8.6	M	Y	A
Device identities	8.7	M	Y	B
Contactless functionality state	8.92	M	Y	C

## 6.6.42 COMMAND CONTAINER

For a non-encrypted encapsulated command, the command format is as follows:

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D+E+F <sub>1</sub> ... F <sub>N</sub> +G)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
eCAT sequence number	8.98	C	N	C
Command details of encapsulated command	8.6	M	Y	D
Device Identities of encapsulated command	8.7	M	Y	E
Other comprehension TLVs of encapsulated command	...	...	...	F <sub>1</sub> ... F <sub>N</sub>
MAC	8.100	C	N	G

eCAT sequence number and MAC shall only be present if the encapsulated command session uses integrity protection only.

For an encrypted encapsulated command, the command format is as follows:

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C+D)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
Encrypted TLV list	8.99	M	Y	C
MAC	8.100	M	N	D

## 6.6.43 ENCAPSULATED SESSION CONTROL

Description	Clause	M/O/C	Min	Length
Proactive UICC command Tag	9.2	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Command details	8.6	M	Y	A
Device Identities	8.7	M	Y	B
SA template	8.101	C	N	C

In case of a request Master SA setup, a request Connection SA setup and a request Secure Channel Start, the SA template comprehension TLV object shall be present; in all other cases, this object shall not be present.

## 6.7 Command results

Once the terminal has made its attempt to execute a proactive command from the UICC, the terminal shall inform the UICC of the success or otherwise of that command, by using TERMINAL RESPONSE. This message gives the command details, including the number of the command (see clause 6.5.1), a general result, and sometimes more specific information.

Three overall categories of results are defined:

- command performed successfully. This is returned by the terminal for every successful command;
- temporary problem with executing command. These are further defined below, but generally these indicate to the UICC that it is worth trying again later;
- permanent problem with executing command. These are further defined below, but generally indicate that the same command will end in the same result if repeated during the same card session.

Successful commands are further defined as:

- command performed successfully. There were no problems;
- command performed with partial comprehension. Here the terminal receives a command with one or more COMPREHENSION-TLV data objects that are unrecognized or unexpected, all of which do not have their "comprehension required" flag set (clause 9.3), but the parent BER-TLV data object still has the minimum set of COMPREHENSION-TLV data objects required to perform the command;
- command performed, with missing information. The terminal received at least the minimum set of component parts, but did not receive all of the parts that it believed mandatory for the UICC to send;
- REFRESH performed with additional eFs read (see clause 6.4.7);
- command performed successfully but requested icon could not be displayed;
- command performed, but modified by call control. This is sent by the terminal to indicate that call control modified the type of request indicated in the proactive command, and that the action requested by call control was performed successfully;
- command performed with modification. This is sent by the terminal to indicate that it is unable to process the command using the exact parameters provided by the UICC. The command is processed with the best possible parameters;
- command performed successfully, limited service;
- REFRESH performed but indicated NAA was not active.

Temporary problems are further defined as:

- terminal is currently unable to process the command. Specific causes for this are:
  - the screen is busy;
  - terminal currently busy on a call;
  - terminal currently busy on SEND DTMF operation;
  - no service is currently available;
  - access control class barred on serving network;
  - no radio resource currently available;
  - not in speech call;
  - no NAA active;
- if none of these can be made to apply, a "no cause can be given" value can be used;
- network is currently unable to process the command. Specific cause values may additionally be provided;
- in some proactive commands, the terminal is required to solicit and receive approval of the user before executing the proactive command. In the case that the user does not give approval for the execution of the proactive command, it shall not be executed by the terminal and the terminal response "user did not accept the proactive command" shall be returned by the terminal to the UICC;
- the user cleared down the call, before the call connected or before the network released the call;
- action in contradiction with the current timer state. This is where the UICC requests an action for a timer to be taken by the terminal and the state of the timer does not allow that action;
- interaction with call control by UICC, temporary problem. This is sent by the terminal to indicate that call control modified the type of request indicated in the proactive command, and that the action requested by call control encounters a temporary problem.

Permanent problems are further defined as:

- command is beyond terminal's capabilities. This is sent by the terminal when it understands what the UICC is asking it to do, but does not have the capability to do it, e.g. terminal which only supports SMS asked to set up a call;
- command type not understood by terminal. This is sent by the terminal when the UICC sends a command with the Type of Command byte set to a value the terminal does not know. This is to allow future expansion of commands;
- command data not understood by terminal. This is sent by the terminal when the command type is understood by the terminal, but the related data object(s) are not, e.g. reserved values have been included in a data object, or one or more unknown COMPREHENSION-TLV data objects have a "comprehension required" tag;
- error, required values are missing. This is given when the command type is understood by the terminal, but it does not receive the minimum set of COMPREHENSION-TLV data objects that it requires to perform the command. These components are shown by the "Min" column in the command structure definitions;
- interaction with call control by NAA by NAA, permanent problem. This is sent by the terminal to indicate that:
  - call control by NAA does not allow the action corresponding to the proactive command; or
  - call control by NAA has modified the type of request indicated in the proactive command and that the action requested by call control encounters a permanent problem;
- specific cause values for this are:
  - action not allowed;
  - the type of request has changed;
  - Current Access Technology unable to process command. This is given to the NAA when terminal is unable to process the requested command due to the current access technology in use;
- if none of these can be made to apply, a "no cause can be given" value can be used.

## 6.8 Structure of TERMINAL RESPONSE

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform. Length (A + B + ... + AB) is indicated by P3 of the header.

Command parameters/data for a TERMINAL RESPONSE to a non-encapsulated command.

Description	Clause	M/O/C	Min	Length
Command details	8.6	M	Y	A
Device identities	8.7	M	N	B
Result	8.12	M	Y	C
Duration (only required in response to a POLL INTERVAL proactive command)	8.8	C	N	D
Text string (only required in response to a GET INKEY or GET INPUT proactive command)	8.15	C	N	E
Item identifier (only required in response to SELECT ITEM proactive command)	8.10	C	N	F

Description	Clause	M/O/C	Min	Length
Local information (only required in response to PROVIDE LOCAL INFORMATION proactive command)	8.19, 8.20, 8.22, 8.29, 8.39, 8.45, 8.46, 8.61, 8.63, 8.74, 8.75, 8.76, 8.81, 8.90	C	N	G
Call control requested action (only required if call control by NAA has modified a proactive command SET UP CALL in another type of request)	8.30	C	N	H
Result data object 2 (only required if call control by NAA has modified a proactive command SET UPSET-UP CALL in another type of request)	8.12	C	N	I
Card reader status (only required in response to GET READER STATUS command). According to the requested information, one Card reader status object for each card interface reported, or one Card reader identifier object is required	8.33, 8.57	C	N	$J_0 + \dots + J_n$ , or J
Card ATR (only required in response to POWER ON CARD)	8.34	C	N	K
R-APDU (only required in response to PERFORM CARD APDU)	8.36	C	N	L
Timer identifier (only required in response to a TIMER MANAGEMENT proactive command)	8.37	C	N	M
Timer value (only required in response to a TIMER MANAGEMENT proactive command)	8.38	C	N	N
AT Response (only required in response to RUN AT COMMAND proactive command)	8.41	C	N	P
Channel data (only required in response to RECEIVE DATA)	8.53	C	N	R
Channel status (only required in response to GET CHANNEL STATUS or OPEN CHANNEL proactive command)	8.56	C	N	$S_0 + \dots + S_n$
Channel data length (only required in response to RECEIVE DATA or SEND DATA proactive command)	8.54	C	N	T
Bearer description (only required in response to OPEN CHANNEL proactive commands, where Bearer description is mandatory in the command)	8.52	C	N	U
Buffer size (only required in response to OPEN CHANNEL proactive command)	8.55	C	N	V
Total display duration (only required in response to a GET INKEY proactive command)	8.8	C	N	W
Service availability (only required in response to SERVICE SEARCH proactive command)	8.67	C	N	X
Service record (only required in response to GET SERVICE INFORMATION proactive command)	8.63	C	N	Y
Other address (local address) (only required in response to OPEN CHANNEL proactive command with dynamic local address request)	8.58	C	N	Z
Frames Information (only required in response to SET FRAMES or GET FRAMES STATUS proactive commands)	8.79	C	N	AA
SA template (only required in response to ENCAPSULATED SESSION CONTROL - request Master SA setup, ENCAPSULATED SESSION CONTROL - request Connection SA setup and ENCAPSULATED SESSION CONTROL - request Secure Channel Start)	8.101	C	N	AB

Command parameters/data for a TERMINAL RESPONSE to a non-encrypted encapsulated command.

Description	Clause	M/O/C	Min	Length
Command details	8.6	M	Y	A
Device identities	8.7	M	N	B
Result	8.12	M	Y	C
eCAT sequence number	8.98	C	N	D
Command details of encapsulated command	8.6	M	Y	E
Device identities of encapsulated command	8.7	M	N	F
Result of encapsulated command	8.12	M	Y	G
Other comprehension TLVs of encapsulated terminal response	...	...	...	H <sub>1</sub> ... H <sub>N</sub>
MAC	8.100	C	N	I

eCAT sequence number and MAC shall only be present if the encapsulated command session uses integrity protection only.

Command parameters/data for a TERMINAL RESPONSE to an encrypted encapsulated command.

Description	Clause	M/O/C	Min	Length
Command details	8.6	M	Y	A
Device identities	8.7	M	N	B
Result	8.12	M	Y	C
Encrypted TLV list	8.99	M	Y	D
MAC	8.100	M	N	E

Under no circumstances shall the UICC wait indefinitely for a TERMINAL RESPONSE.

For all the Conditional (C) COMPREHENSION-TLV objects, the terminal should not include them in the response to non-applicable situations. However, if one is present, the UICC shall ignore it.

For all COMPREHENSION-TLV objects with Min = N, the terminal should set the CR flag to comprehension not required. Any future additional COMPREHENSION-TLV objects will be included as Min = N and comprehension not required. This will ensure that any proactive command will end in a predictable way.

Response parameters/data: None.

## 6.8.1 Command details

This data object shall be identical to the command details data object (including the comprehension required flag) given by the UICC in the proactive command to which the terminal is giving the result:

- if the terminal has not received a valid Command number, all Command Details object values shall be set to '00' and the Result shall indicate an error;
- if the failure is caused by a problem on the transmission layer, the terminal shall respond with "temporary problem" ("terminal currently not able to process command"). If not, the terminal shall respond with "permanent problem" (either "command not understood by terminal" or "Error required values are missing");
- the UICC shall interpret a terminal Response with a command number '00' as belonging to the last proactive command having been sent to the terminal.

## 6.8.2 Device identities

The terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

### 6.8.3 Result

This data object holds the result of the proactive UICC command.

### 6.8.4 Duration

When the terminal issues a successful TERMINAL RESPONSE for a POLL INTERVAL command, it shall state the polling interval it will be using in the Duration data object.

### 6.8.5 Text string

When the terminal issues a successful TERMINAL RESPONSE ('0X' result value - refer to clause 8.12) for a GET INKEY or GET INPUT command, it shall supply the single character or the character string entered by the user in the Text string data object. When the terminal issues a successful TERMINAL RESPONSE ("0X" result value - refer to clause 8.12) for a GET INKEY ("Yes/No") command with command qualifier set to "Yes/No", it shall supply the value "01" when the answer is "positive" and the value "00" when the answer is "negative" in the text string data object.

When the terminal issues a successful TERMINAL RESPONSE ('0X' result value - refer to clause 8.12) for a GET INPUT command to which the user has made an empty input (i.e. if the user does not enter any character), the terminal shall indicate this by means of either a null text string (see clause 8.15 for the coding of this object), or by means of a Text string object with Length = '01', and a Value part consisting of a data coding scheme only.

NOTE: The notion of empty input is different from the general result "no response from user" (see clause 8.12). The latter event is typically caused by a timeout in the MMI, whereas an empty input requires an acknowledgement from the user.

### 6.8.6 Item identifier

When the terminal issues a successful TERMINAL RESPONSE ('0X' result value - refer to clause 8.12) for a SELECT ITEM command, it shall supply the identifier of the item selected by the user in the Item identifier data object. If the terminal issues a TERMINAL RESPONSE with result "Help information required by the user" for a SELECT ITEM command, it shall supply the identifier of the item for which the user is requiring help information.

### 6.8.7 Local information

When the terminal issues a successful TERMINAL RESPONSE for a PROVIDE LOCAL INFORMATION command, it shall supply the requested local information:

- Where the UICC has requested location information, TERMINAL RESPONSE shall contain the location information data object.
- Where the UICC has requested location information for multiple access technologies, TERMINAL RESPONSE shall contain the Access Technology data object listing all current access technologies, followed by one location information data object for each current access technology in the same sequence. If no location information is available for an access technology, the respective data object shall have length zero.
- Where the UICC has requested the IMEI, TERMINAL RESPONSE shall contain the IMEI data object.
- Where the UICC has requested the Network Measurement Results the TERMINAL RESPONSE shall contain the NMR data object and the BCCH channel list data object if supported by the network access technology.
- Where the UICC has requested the Network Measurement Results for multiple access technologies, TERMINAL RESPONSE shall contain the Access Technology data object listing all current access technologies, followed by one NMR data object and one BCCH channel list data object for each current access technology in the same sequence. The BCCH channel list data object shall immediately follow the NMR data object, even if not supported by a network access technology. If no NMR data or no BCCH channel list is available for an access technology, the respective data object shall have length zero.
- Where the UICC has requested the date, time and time zone the TERMINAL RESPONSE shall contain the Date-Time and Time zone data object.



- Where the UICC has requested the currently used language, the TERMINAL RESPONSE shall contain the Language data object.
- Where the UICC has requested the Timing Advance, the TERMINAL RESPONSE shall contain the Timing Advance data object if supported by the network access technology.
- Where the UICC has requested the Battery State, the TERMINAL RESPONSE shall contain the Battery State data object (if class "g" is supported).
- Where the UICC has requested the Access Technology (single access technology), the TERMINAL RESPONSE shall contain the Access Technology data object with length set to 1. If the terminal is currently connected via more than one technology, it shall select the most appropriate value.
- Where the UICC has requested Multiple Access Technologies, the TERMINAL RESPONSE shall include all access technologies that the terminal is currently connected to.
- Where the UICC has requested the ESN, TERMINAL RESPONSE shall contain the ESN data object.
- Where the UICC has requested the IMEISV, TERMINAL RESPONSE shall contain the IMEISV data object.
- Where the UICC has requested the Search Mode information, the TERMINAL RESPONSE shall contain the Search Mode data object.
- Where the UICC has requested the MEID, TERMINAL RESPONSE shall contain the MEID data object.
- Where the UICC has requested broadcast network information, TERMINAL RESPONSE shall contain the broadcast network information data object (if class "o" is supported).

### 6.8.8 Call control requested action

When the terminal issues a TERMINAL RESPONSE for a proactive command SET UP CALL which has been modified by call control by UICC in another type of request, it shall supply the response data given in response to the ENVELOPE (CALL CONTROL).

### 6.8.9 Result data object 2

When the terminal issues a TERMINAL RESPONSE for a proactive command SET UP CALL which has been modified by call control by UICC in another type of request, it shall supply the Result data object it would have supplied for the proactive command equivalent to the action requested by call control, and given in the Call control request data element.

### 6.8.10 Card reader status

This clause applies if class "a" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a GET READER STATUS command, it shall supply the requested readers' information:

- Where the UICC has requested the card reader status, TERMINAL RESPONSE shall supply the status of each card reader in n consecutive Card reader status data objects, where n is the card reader count.
- Where the UICC has requested the card reader identifier, TERMINAL RESPONSE shall supply the identifier of the requested card reader identifier.

### 6.8.11 Card ATR

This clause applies if class "a" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a POWER ON CARD command, it shall supply the ATR returned by the addressed card in the Card ATR data object.

## 6.8.12 R-APDU

This clause applies if class "a" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a PERFORM CARD APDU command, it shall supply the response data and status words in the R-APDU data object.

## 6.8.13 Timer identifier

When the terminal issues a successful TERMINAL RESPONSE for a TIMER MANAGEMENT, it shall state in the timer identifier data object the identifier of the timer to which this command applies.

## 6.8.14 Timer value

When the terminal issues a successful TERMINAL RESPONSE for a TIMER MANAGEMENT command with command qualifier indicating "deactivate" or "get the current value of the timer", it shall state in the timer value data object the current value of the timer.

## 6.8.15 AT Response

This clause applies if class "b" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a RUN AT COMMAND command, the TERMINAL RESPONSE shall contain the AT Response (as defined in clause 8.40).

## 6.8.16 Text string 2

The presence of this object is access technology dependant.

## 6.8.17 Channel data

This clause applies if class "e" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a RECEIVE DATA command, the TERMINAL RESPONSE shall contain the Channel Data data object.

## 6.8.18 Channel status

This clause applies if class "e" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a GET CHANNEL STATUS proactive command, the TERMINAL RESPONSE shall contain as many Channel Status data objects as there are available channels.

When the terminal issues a successful TERMINAL RESPONSE for an OPEN CHANNEL command, the TERMINAL RESPONSE shall contain a Channel status data object for the opened channel.

## 6.8.19 Channel data length

This clause applies if class "e" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a RECEIVE DATA command or a SEND DATA, the TERMINAL RESPONSE shall contain the Channel Data Length data object.

## 6.8.20 Bearer description

This clause applies if class "e" is supported.

When the terminal issues a successful or an unsuccessful TERMINAL RESPONSE for an OPEN CHANNEL command, the TERMINAL RESPONSE shall contain the Bearer description data object.

### 6.8.21 Buffer size

This clause applies if class "e" is supported.

When the terminal issues a successful or an unsuccessful TERMINAL RESPONSE for an OPEN CHANNEL command, the TERMINAL RESPONSE shall contain the Buffer size data object.

### 6.8.22 Total display duration

When the terminal issues a TERMINAL RESPONSE for a GET INKEY proactive command with variable timeout, it shall supply the total display text duration (*command execution duration*). The time unit of the response is identical to the time unit of the requested variable timeout.

Resolution and the precision of the time value are in accordance with clause 6.4.21.

### 6.8.23 Service Availability

This clause applies if class "f" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a SERVICE SEARCH command, the TERMINAL RESPONSE shall contain the Service Availability data object.

### 6.8.24 Service Record

This clause applies if class "f" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a GET SERVICE INFORMATION command, the TERMINAL RESPONSE shall contain the Service Record data object.

### 6.8.25 Other address (local address)

This clause applies if class "e" is supported.

When the terminal issues a successful TERMINAL RESPONSE for an OPEN CHANNEL command with dynamic local address request, the TERMINAL RESPONSE shall contain an Other address data object for the opened channel.

### 6.8.26 Frames Information

This clause applies if class "i" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a SET FRAMES proactive command, the TERMINAL RESPONSE shall contain as many frames information data objects as there are frames required.

Frames information data objects shall match with frames from left to right and top to bottom (see annex R).

### 6.8.27 SA template

This clause applies if class "x" is supported.

When the terminal issues a successful TERMINAL RESPONSE for an ENCAPSULATED SESSION CONTROL command requesting Master SA setup, Connection SA setup or Secure Channel Start, the TERMINAL RESPONSE shall contain the SA template data object.

## 6.8.28 eCAT sequence number

This clause applies if class "x" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a COMMAND CONTAINER command and the encapsulated command session is integrity protected only, the TERMINAL RESPONSE shall contain the eCAT sequence number data object.

## 6.8.29 Encrypted TLV list

This clause applies if class "x" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a COMMAND CONTAINER command and the encapsulated command session is encrypted, the TERMINAL RESPONSE shall contain the encrypted TLV list data object.

## 6.8.30 MAC

This clause applies if class "x" is supported.

When the terminal issues a successful TERMINAL RESPONSE for a COMMAND CONTAINER command and the encapsulated command session is integrity protected, the TERMINAL RESPONSE shall contain the MAC data object.

# 6.9 Proactive UICC session and terminal display interaction

During a proactive session the terminal display shall be refreshed by any display data contained in the first and each subsequent proactive command. The refresh shall occur once the terminal has retrieved the proactive command using the Fetch instruction, following the proactive command pending status response.

If no proactive command is pending (status response of '90 00' following the terminal Response), then the session releases the display back into terminal control. If this session was terminated in a backwards move, and the session was initiated from an Envelope command containing a Menu Selection, it is recommended that the display returns to the Setup Menu.

If the text is to be sustained, the terminal shall display the text of applicable DISPLAY TEXT commands beyond the sending of the TERMINAL RESPONSE and possibly beyond the end of the proactive session.

If a variable display timeout was indicated for a DISPLAY TEXT command, then the session releases the display back into terminal control no later than the period stated by the duration. If the text is to be sustained beyond an immediate response, the terminal shall display the text for a period that does not exceed the duration.

The procedure described above applies also for DISPLAY MULTIMEDIA MESSAGE.

# 6.10 Handling of unknown, unforeseen and erroneous messages

## 6.10.1 General

The procedures described in this clause apply to the BER-TLV and COMPREHENSION-TLV data objects described in the present document. The purpose of this clause is to allow greater flexibility in future versions of the present document, and a greater predictability across different versions of the present document.

The procedures described here specify how the terminal and UICC shall behave when they receive a proactive command or response that is not fully compliant with the standards by which it was designed. A response will be made to the UICC by means of the "general result" field of the "result".

If the terminal sends a FETCH or TERMINAL RESPONSE to the UICC that contains values that the UICC does not understand, then the UICC shall issue the appropriate SW1/SW2 error response. The current proactive transaction shall be considered complete and neither the terminal nor the UICC shall take any further action with regard to it. In this case, unless the "General result" is "command performed..." then the UICC shall assume that the command was not carried out and that a permanent error exists with regard to that particular proactive command. If the command was performed, but the "additional information on result" field was not understood, then the UICC may attempt the command again at a later stage in the current card session.

If the UICC has enough information to proceed (i.e. it has received all the data objects of the Minimum set) then it shall do so.

### 6.10.2 Message too short

Any information received that is not a complete tag and length shall be ignored.

### 6.10.3 Missing minimum information

If a message is received that does not have all the mandatory elements in it, then if all of the minimum set elements are present then the receiver shall complete the command and report "command performed, with missing information".

If the minimum set of elements is not complete, then the terminal shall respond with "Error, required values are missing".

### 6.10.4 Unknown Tag value

If a BER-TLV object is received that has a tag that is understood, but contains COMPREHENSION-TLV components that have unknown tags, then provided the minimum set condition is fulfilled, the "comprehension required" bit of the tag shall determine how the receiving entity behaves.

If the comprehension required flag in an unknown tag is set to '1', and the terminal either does not recognize or is not expecting one or more of the COMPREHENSION-TLV objects in the message, then it shall respond with "Command data not understood by terminal".

If the comprehension required flag is set to "0", then the terminal shall read the length field that follows and ignore that object. In this case the terminal will be able to carry out the command without the COMPREHENSION-TLV components that it cannot understand. It shall respond with "command performed with partial comprehension".

### 6.10.5 Unexpected Tag value

If a BER-TLV object is received that contains elements that have recognizable tags, but which were not expected in the context of this message (for example, the terminal sees SMS TPDU tag as part of DISPLAY TEXT), then it shall discard that element. It shall then proceed as described for Unknown Tag values.

If a received object has a tag that has already been received, then the first instance shall be used and any subsequent instances shall be discarded.

### 6.10.6 Length errors

If the total lengths of the COMPREHENSION-TLV data objects are not consistent with the length given in the BER-TLV data object, then the whole BER-TLV data object shall be rejected. The result field in the TERMINAL RESPONSE shall have the error condition "Command data not understood by terminal".

If the length of the BER-TLV data object is shorter than the length of the response data, the terminal shall ignore response data following the complete BER-TLV data object. If the length of the BER-TLV data object is longer than the length of the response data, then clauses 6.10.2 and 6.10.3 apply.

## 6.10.7 Contents not understood

If the contents of a COMPREHENSION-TLV data object contain a field with a value that is defined as reserved, then the whole COMPREHENSION-TLV data object shall be considered as invalid. It will then depend on the "comprehension required" bit of the relevant tag as to whether the whole BER-TLV data object shall be rejected, or whether that particular COMPREHENSION-TLV data object shall be ignored.

If the contents of a BER-TLV object contain RFU bits or bytes, then these shall be ignored.

## 6.10.8 Extended length data objects

If a COMPREHENSION-TLV data object has a length longer than expected (i.e. more information has been added), then the receiver shall ignore this extra information to the end of the object. The end of the object shall be found by looking at the "length" field of that object.

NOTE: If comprehension of the extra bytes is required, this can be achieved by the use of a reserved coding in an earlier field.

## 6.11 Proactive commands versus possible terminal response

Tables 6.1 and 6.2 show for each proactive command the possible terminal response returned (marked by a "•" character).

**Table 6.1: Proactive commands versus possible terminal response (continued overleaf)**

		PROACTIVE COMMAND																
		RE-FRESH	MORE TIME	POLL INTERVAL	POLLING OFF	SETUP EVENT LIST	SET UP CALL	SEND DTMF	LAUNCH BROWSER	PLAY TONE	DISPLAY TEXT	GET INKEY	GET INPUT	SELECT ITEM	SET UP MENU	PROVIDE LOCAL INFO	TIMER MANAGEMENT	SET-UP IDLE MODE TEXT
TERMINAL RESPONSE		'01'	'02'	'03'	'04'	'05'	'10'	'14'	'15'	'20'	'21'	'22'	'23'	'24'	'25'	'26'	'27'	'28'
00	Command performed successfully	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
01	Command performed with partial comprehension	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
02	Command performed, with missing information	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
03	REFRESH performed with additional EFs read	•																
04	Command performed successfully, but requested icon could not be displayed	•					•	•		•	•	•	•	•	•			
05	Command performed, but modified by call control by NAA						•											
06	Command performed successfully, limited service															•		
07	Command performed with modification																	
08	REFRESH performed but indicated NAA was not active	•																
09	Command performed successfully, tone not played									•								
10	Proactive UICC session terminated by the user						•	•		•	•	•	•	•				
11	Backward move in the proactive UICC session requested by the user										•	•	•	•				
12	No response from user										•	•	•	•				
13	Help information required by the user										•	•	•					
14	Reserved for 3GPP						•											
20	Terminal currently unable to process command	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
21	Network currently unable to process command						•		•									
22	User did not accept the proactive command						•											
23	User cleared down call before connection or network release						•											
24	Action in contradiction with the current timer state																•	
25	Interaction with call control by NAA, temporary problem						•											
26	Launch browser generic error								•									
27	MMS Temporary Problem																	
30	Command beyond terminal's capabilities	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
31	Command type not understood by terminal	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
32	Command data not understood by terminal	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
33	Command number not known by terminal	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
34	Reserved for 3GPP						•											
35	Reserved for 3GPP																	
36	Error, required values are missing	•	•	•	•	•	•	•		•	•	•	•	•	•	•	•	•
37	Reserved for 3GPP																	
38	Multiple Card command error																	
39	Interaction with call control by NAA, permanent problem						•											
3A	Bearer Independent Protocol error																	

		PROACTIVE COMMAND																
		RE-FRESH	MORE TIME	POLL INTERVAL	POLLING OFF	SETUP EVENT LIST	SET UP CALL	SEND DTMF	LAUNCH BROWSER	PLAY TONE	DISPLAY TEXT	GET INKEY	GET INPUT	SELECT ITEM	SET UP MENU	PROVIDE LOCAL INFO	TIMER MANAGEMENT	SET-UP IDLE MODE TEXT
TERMINAL RESPONSE		'01'	'02'	'03'	'04'	'05'	'10'	'14'	'15'	'20'	'21'	'22'	'23'	'24'	'25'	'26'	'27'	'28'
3B	Access Technology unable to process command																	
3C	Frames error	•					•	•	•	•	•	•	•	•				•
3D	MMS error																	

Table 6.2: Proactive commands versus possible terminal response (continued overleaf)

		PROACTIVE COMMAND																			
		CARD APDU	POWER ON CARD	POWER OFF CARD	GET READER STATUS	RUN AT COMMAND	LANGUAGE NOTIFICATION	OPEN CHANNEL	CLOSE CHANNEL	RECEIVE DATA	SEND DATA	GET CHANNEL STATUS	SERVICE SEARCH	GET SERVICE INFORMATION	DECLARE SERVICE	SET FRAMES	GET FRAMES STATUS	RETRIEVE MM	SUBMIT MM	DISPLAY MM	ACTIVATE
TERMINAL RESPONSE		'30'	'31'	'32'	'33'	'34'	'35'	'40'	'41'	'42'	'43'	'44'	'45'	'46'	'47'	'50'	'51'	'60'	'61'	'62'	'70'
00	Command performed successfully	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
01	Command performed with partial comprehension	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
02	Command performed, with missing information	•	•	•	•	•	•	•	•	•	•	•			•			•	•	•	
03	REFRESH performed with additional EFs read																				
04	Command performed successfully, but requested icon could not be displayed							•	•	•	•	•	•								
05	Command performed, but modified by call control by NAA																				
06	Command performed successfully, limited service																				
07	Command performed with modification							•							•						
08	REFRESH performed but indicated NAA was not active							•													
09	Command performed successfully, tone not played																				
10	Proactive UICC session terminated by the user							•	•	•	•	•	•							•	
11	Backward move in the proactive UICC session requested by the user																				
12	No response from user																				
13	Help information required by the user																				
14	Reserved for 3GPP																				
20	Terminal currently unable to process command	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•				•
21	Network currently unable to process command							•			•										
22	User did not accept the proactive command							•			•										
23	User cleared down call before connection or network release																				
24	Action in contradiction with the current timer state																				
25	Interaction with call control by NAA, temporary problem							•													
26	Launch browser generic error																				
27	MMS Temporary Problem																	•	•	•	
30	Command beyond terminal's capabilities	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•



		PROACTIVE COMMAND																			
		CARD APDU	POWER ON CARD	POWER OFF CARD	GET READER STATUS	RUN AT COMMAND	LANGUAGE NOTIFICATION	OPEN CHANNEL	CLOSE CHANNEL	RECEIVE DATA	SEND DATA	GET CHANNEL STATUS	SERVICE SEARCH	GET SERVICE INFORMATION	DECLARE SERVICE	SET FRAMES	GET FRAMES STATUS	RETRIEVE MM	SUBMIT MM	DISPLAY MM	ACTIVATE
<b>TERMINAL RESPONSE</b>		'30'	'31'	'32'	'33'	'34'	'35'	'40'	'41'	'42'	'43'	'44'	'45'	'46'	'47'	'50'	'51'	'60'	'61'	'62'	'70'
31	Command type not understood by terminal	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
32	Command data not understood by terminal	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
33	Command number not known by terminal	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
34	Reserved for 3GPP																				
35	Reserved for 3GPP																				
36	Error, required values are missing	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•	•
37	Reserved for 3GPP																				
38	Multiple Card command error	•	•	•	•																
39	Interaction with call/SM control by NAA, permanent problem																				
3A	Bearer Independent Protocol error							•	•	•	•		•	•	•						
3B	Access Technology unable to process command							•					•	•							
3C	Frames error					•		•	•	•	•		•	•		•	•				
3D	MMS error																	•	•	•	

Table 6.3: Proactive commands versus possible terminal response

		PROACTIVE COMMAND																			
		COMM AND CONTAINER	ENCAPULATED SESSION CONTROL																		
<b>TERMINAL RESPONSE</b>		'72'	'73'																		
00	Command performed successfully	•	•																		
01	Command performed with partial comprehension	•	•																		
02	Command performed, with missing information																				
03	REFRESH performed with additional EFs read																				
04	Command performed successfully, but requested icon could not be displayed																				
05	Command performed, but modified by call control by NAA																				
06	Command performed successfully, limited service																				
07	Command performed with modification																				
08	REFRESH performed but indicated NAA was not active																				
09	Command performed successfully, tone not played																				
10	Proactive UICC session terminated by the user																				

		PROACTIVE COMMAND													
		COMM AND CONTAINER	ENCAPSULATED SESSION CONTROL												
<b>TERMINAL RESPONSE</b>		'72'	'73'												
11	Backward move in the proactive UICC session requested by the user														
12	No response from user														
13	Help information required by the user														
14	Reserved for 3GPP														
20	Terminal currently unable to process command	•	•												
21	Network currently unable to process command														
22	User did not accept the proactive command														
23	User cleared down call before connection or network release														
24	Action in contradiction with the current timer state														
25	Interaction with call control by NAA, temporary problem														
26	Launch browser generic error														
27	MMS Temporary Problem														
30	Command beyond terminal's capabilities	•	•												
31	Command type not understood by terminal	•	•												
32	Command data not understood by terminal	•	•												
33	Command number not known by terminal	•	•												
34	Reserved for 3GPP														
35	Reserved for 3GPP														
36	Error, required values are missing	•	•												
37	Reserved for 3GPP														
38	Multiple Card command error														
39	Interaction with call control by NAA, permanent problem														
3A	Bearer Independent Protocol error														
3B	Access Technology unable to process command														
3C	Frames error														
3D	MMS error														

## 7 ENVELOPE commands

### 7.1 Void

### 7.2 Menu selection

A set of possible menu options can be supplied by the UICC using the proactive command SET UP MENU. If the UICC has sent this command, and the user subsequently chooses an option or, the user requests help on it, the terminal informs the UICC using this procedure.

#### 7.2.1 Procedure

The terminal shall follow the procedure below.

- When the terminal receives a menu selection from one of the menu items defined by a "SET-UP MENU" command issued previously by the UICC, or the user has indicated the need to get help information on one of these menu items, then it shall pass the identifier of the selected menu item to the UICC using the ENVELOPE (MENU SELECTION) command, as defined in clause 7.2.2.
- If the UICC responds with '93 00', the terminal shall not re-issue this particular envelope.

#### 7.2.2 Structure of ENVELOPE (MENU SELECTION)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Menu Selection tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Device identities	8.7	M	Y	A
Item identifier	8.10	M	Y	B
Help request	8.21	O	N	C

Device identities: the terminal shall set the device identities to:

- source: Keypad;
- destination: UICC.

Help request: inclusion of this data object depends upon whether the user actually selected the named menu item or just requested help information on it. If the user actually selected the menu item, this data object shall not be included. If the user indicated the need to get help information on the menu item, this data object shall be included.

Response parameters/data: None for this type of ENVELOPE command.

## 7.3 Call Control by NAA

### 7.3.1 Call Control by NAA

#### 7.3.1.1 Procedure for mobile originated calls

If the service "call control" is available in the Service Table provided by the NAA, then the terminal shall follow the procedure below:

- for all call set-up attempts (even those resulting from a SET UP CALL proactive UICC command, from the Bearer Independent Protocol proactive UICC commands where CSD is selected, or those occurring when another call is already in progress, and those resulting from automatic redial attempts), the terminal shall first pass the call set-up details (dialled digits and associated parameters) to the UICC, using the ENVELOPE (CALL CONTROL) command defined below. The "Location Information" shall be the current information, even for automatic redial attempts. CAT applications should take into account the following exception:
  - when the user is dialling an emergency call code, the terminal sets up an emergency and does not pass the call set-up details to the UICC;
- if the UICC responds with '90 00', the terminal shall set up the call with the dialled digits and other parameters as sent to the UICC;
- if the UICC responds with '93 00', the terminal shall not set up the call and may retry the command;
- if the UICC provides response data, then the response data from the UICC shall indicate to the terminal whether to set up the call as proposed, not set up the call, set up a call using the data supplied by the UICC. It is mandatory for the terminal to perform the call set-up request in accordance with the data from the UICC, if it is within the terminal's capabilities to do so. If the UICC requires a call set-up that is beyond the terminal's capabilities (e.g. the UICC maps a speech call to a data call, and the terminal does not support data calls), then the terminal shall not perform the call set-up request at all. It is possible for the UICC to request the terminal to set up an emergency call by supplying the number "112" as the response data.

In the case where the initial call set-up request results from a proactive command SET UP CALL:

- if the call control result is "not allowed", the terminal shall inform the UICC using TERMINAL RESPONSE "interaction with call control by NAA, permanent problem; action not allowed".

If the terminal supports the Last Number Dialed service, the terminal shall update EF<sub>LND</sub> with the call set-up details (digits string and associated parameters) corresponding to the initial user request.

The terminal shall then follow the call set-up procedure defined in the relevant Access Technology specification.

#### 7.3.1.2 Void

#### 7.3.1.3 Indication to be given to the user

The UICC may optionally include an alpha-identifier in the response data to the ENVELOPE (CALL CONTROL) message, in order to inform the user at the time the response is received by the terminal. The use of this alpha identifier by the terminal is described below:

- if the UICC responds with "allowed, no modification", then:
  - if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it to inform the user during the call set-up;
  - if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not modify the display corresponding to the initial user request;
  - if the alpha identifier is not provided by the UICC, the terminal may give information to the user concerning what is happening;

- if the UICC responds with "not allowed", then:
  - if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it to inform the user. This is also an indication that the terminal should not give any other information to the user on the reason of the barring;
  - if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), the terminal may give information to the user concerning what is happening;
  - if the alpha identifier is not provided by the UICC, the terminal may give information to the user concerning what is happening.
- if the UICC responds with "allowed, with modifications", and the modified request is within the terminal's capabilities, then:
  - if the alpha identifier is provided by the UICC and is not a null data object, the terminal shall use it to inform the user. The terminal shall then not display the destination address given by the UICC. This is also an indication that the terminal should not give any other information to the user on the changes made by the UICC to the initial user request;
  - if the alpha identifier is provided by the UICC and is a null data object (i.e. length = '00' and no value part), this is an indication that the terminal should not give any information to the user on the changes made by the UICC to the initial user request. The terminal shall not display the destination address given by the UICC. The terminal should not modify the display corresponding to the initial user request;
  - if the alpha identifier is not provided by the UICC, the terminal may indicate to the user that the initial user request has been changed;
- if the UICC responds with "allowed, with modifications" to a user-initiated request (i.e. a request not initiated by a proactive command), and the modified user request is beyond the terminal's capabilities, then the terminal may give information to the user on the modified request and the fact that the modified request is beyond the terminal's capabilities, optionally using the alpha identifier, if one is provided by the UICC;
- if the UICC responds with "allowed, with modifications" to a request by a proactive command SET UP CALL, and the modified request is beyond the terminal's capabilities, then the terminal shall not give any information to the user on the fact that the modified request is beyond the terminal's capabilities, and shall give a TERMINAL RESPONSE to the proactive command (i.e. SET UP CALL) as detailed in clause 7.3.1.1. The responsibility to inform the user in this case lies with the UICC application which sent the proactive command.

#### 7.3.1.4 Interaction with Fixed Dialling Number (FDN)

It is permissible for the Fixed Dialling Number (FDN) service to be enabled at the same time as Call Control is available in the NAA Service Table.

If FDN is enabled and Call Control is activated, the terminal shall follow this procedure:

- the terminal shall check that the number entered through the MMI is on the FDN list;
- if the MMI input does not pass the FDN check, the call shall not be set-up;
- if the MMI input does pass the FDN check, the terminal shall pass the dialled digits and other parameters to the UICC, using the ENVELOPE (CALL CONTROL) command;
- if the UICC responds with "allowed, no modification", the terminal shall set up the call as proposed;
- if the UICC responds with "not allowed", the terminal shall not set up the call;
- if the UICC responds with "allowed with modifications", the terminal shall set up the call in accordance with the response from the UICC. If the modifications involve changing the dialled digits, the terminal shall not re-check this modified number (or string) against the FDN list.

If the user wishes to enable or disable Fixed Dialling Number, the terminal shall follow the procedure defined in the relevant access technology specification. The state of the Call Control service shall have no effect on this procedure.

### 7.3.1.5 Support of Barred Dialling Number (BDN) service

The BDN service shall be allocated and activated in the NAA Service Table only if Call Control is also available in the NAA Service Table.

If Barred Dialling Number service is enabled, when receiving the dialled number and other parameters from the terminal, the NAA may check this information against those stored in  $EF_{BDN}$  (examples of comparison methods are given in TS 100 906 [i.1]):

- if the UICC responds with "not allowed" (e.g. a match is made against a BDN), the terminal shall not set up the call;
- if the UICC responds with "allowed, no modification", the terminal shall set up the call as proposed;
- if the UICC responds with "allowed with modifications", the terminal shall set up the call in accordance with the response from the UICC. If the modifications involve changing the dialled number, the terminal shall not re-check this modified number (or string) against the FDN list when FDN is enabled.

If the user wishes to enable or disable Barred Dialling Number, the terminal shall follow the procedure defined in the relevant access technology specification.

### 7.3.1.6 Structure of ENVELOPE (CALL CONTROL)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Call control tag	9.1	M	Y	1
Length (A+B+C+D+E+F+G)	-	M	Y	1 or 2
Device identities	8.7	M	Y	A
Address	8.1	M	Y	B
Capability configuration parameters 1	8.4	O	N	C
Subaddress	8.3	O	N	D
Location information	8.19	M	N	E
Capability configuration parameters 2	8.4	O	N	F
BC repeat indicator	8.42	O	N	G

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Address: only one data object shall be sent to the UICC:

- for a call set-up, the address data object is used and holds the Called Party Number, to which the terminal is proposing setting up the call.

Capability configuration parameters: only used for a call set-up, this contains the Bearer capabilities that the terminal is proposing to send to the network. The first capability configuration parameters correspond to the bearer capability 1 information element of a mobile originating call setup message. The second capability configuration parameters correspond to the bearer capability 2 information element of a mobile originating call setup message. If no capability configuration parameters are present, this shall indicate a speech call.

BC repeat indicator: indicates how the 2 associated bearers shall be interpreted. This BC repeat indicator is optional if the second capability configuration parameter is present. It shall not be present if the second capability configuration parameter is not present.

Subaddress: only used for a call set-up, this contains the called party subaddress that the terminal is proposing to send to the network. If one is not present, this shall indicate that the terminal is proposing not to send this information element to the network.

Location information: this data object contains the identification of the current serving cell of the terminal. The comprehension required flag of this data object in this command shall be set to '0'.

Response parameters/data for a non-encapsulated envelope.

It is permissible for the UICC to provide no response data, by responding with SW1/SW2 = "90 00". If the UICC does not provide any response data, then this shall have the same meaning as "allowed, no modification".

Description	Clause	M/O/C	Min	Length
Call control result	-	M	Y	1
Length (A+B+C+D+E+F)	-	M	Y	1 or 2
Address	8.1	O	N	A
Capability configuration parameters 1	8.4	O	N	B
Subaddress	8.3	O	N	C
Alpha identifier	8.2	O	N	D
BC repeat indicator	8.42	C	N	E
Capability configuration parameters 2	8.4	O	N	F

Call control result:

- Contents:
  - the command that the UICC gives to the terminal concerning whether to allow, bar or modify the proposed call.
- Coding:
  - '00' = Allowed, no modification;
  - '01' = Not allowed;
  - '02' = Allowed with modifications.

Address: only one data object may be included if the UICC requests the call details to be modified:

- for a call set-up, if the address data object is not present, then the terminal shall assume the Dialling number is not to be modified.

Capability configuration parameters: only used for a call set-up, this data object is only required if the NAA application requests the call details to be modified. The first capability configuration parameters correspond to the bearer capability 1 information element of a mobile originating call setup message. The second capability configuration parameters correspond to the bearer capability 2 information element of a mobile originating call setup message. If the capability configuration parameters are not present, then the terminal shall assume the parameters are not to be modified.

Subaddress: only used for a call set-up, this data object is only required if the NAA application requests the call details to be modified. If the subaddress is not present, then the terminal shall assume the called party subaddress is not to be modified. If the subaddress supplied by the NAA application is a null data object, then the terminal shall not provide a called party subaddress to the network. A null data object shall have length = '00' and no value part.

Alpha identifier: this data object is only required if the UICC requests a particular indication to be given to the user. The handling of this data object by the terminal is described in clause 7.3.1.3. The comprehension required flag of this data object shall be set to '0'.

BC repeat indicator: indicates how the 2 associated bearers shall be interpreted. This BC repeat indicator is conditioned to the presence of the second capability configuration parameters.

It is mandatory for the UICC to provide at least one of the optional data objects if it has set the Call control result to "allowed with modifications".

NOTE: The technology specific toolkit specification will define the appropriate call setup message.

Response parameters/data if the call control envelope is encapsulated in a non-secured envelope container (see clause 7.9).

Description	Clause	M/O/C	Min	Length
Call control result	8.97	M	Y	A
Address	8.1	O	N	B
Capability configuration parameters 1	8.4	O	N	C
Subaddress	8.3	O	N	D
Alpha identifier	8.2	O	N	E
BC repeat indicator	8.42	C	N	F
Capability configuration parameters 2	8.4	O	N	G

In this case, the result is a list of comprehension TLV objects. Call control result is mandatory.

## 7.4 Timer expiration

### 7.4.1 Description

When a timer previously started by a **TIMER MANAGEMENT** proactive command expires, the terminal shall pass the identifier of the timer that has expired and its value using the **ENVELOPE (TIMER EXPIRATION)** command, as defined in clause 7.4.2.

If the UICC is busy and returns status '93 00', the terminal shall retry until the command is accepted.

**NOTE:** In order to avoid retrying periodically, the terminal could retry after a **TERMINAL RESPONSE** processed by the UICC with status '90 00'.

### 7.4.2 Structure of ENVELOPE (TIMER EXPIRATION)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Timer Expiration tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Device identities	8.7	M	Y	A
Timer identifier	8.37	M	Y	B
Timer value	8.38	M	Y	C

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Timer identifier: identifier of the timer that has expired.

Timer value: difference between the time when this command is issued and the time when the timer was initially started. This should be as close as possible to the value of the timer given in the initial **TIMER MANAGEMENT** command.

Response parameters/data:

- none.



## 7.5 Event download

A set of events for the terminal to monitor can be supplied by the UICC using the proactive command SET UP EVENT LIST. If the UICC has sent this command, and an event which is part of the list subsequently occurs, the terminal informs the UICC using the procedure below, relevant for that event.

Processing within the terminal resulting from this event shall proceed as normal, independent of sending the ENVELOPE command to the UICC.

Where events occur while the UICC-terminal interface is already busy, the terminal shall queue events and send event download messages to the UICC in the order in which they occurred.

Where events occur and the UICC responds with '93 00', the terminal shall retry to deliver the event download messages to the UICC.

### 7.5.1 MT call event

#### 7.5.1.1 Procedure

If the MT call event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the terminal receives an incoming call setup message, the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - MT call) command as defined in clause 7.5.1.2.

NOTE: The technology specific toolkit specification will define the appropriate call setup message.

#### 7.5.1.2 Structure of ENVELOPE (EVENT DOWNLOAD - MT call)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C+D+E)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Transaction identifier	8.28	M	Y	C
Address	8.1	C	N	D
Subaddress	8.3	C	N	E

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- MT call.

Device identities: the terminal shall set the device identities to:

- source: network;
- destination: UICC.

Transaction identifier: the transaction identifier data object shall contain one transaction identifier, and this shall be the Transaction Identifier in the call setup message from the network.

Address: the address data object holds the Calling Party number received by the terminal in the call setup message. If the Calling Party number is included in the call setup message, the terminal shall include the Address object, otherwise the terminal shall not include the Address object.

Subaddress: The Subaddress data object holds the Calling Party Subaddress as received by the terminal in the call setup message. If the Calling Party Subaddress is included in the call setup message, the terminal shall include the Subaddress object, otherwise the terminal shall not include the Subaddress object.

Response parameters/data:

- none.

## 7.5.2 Call connected event

### 7.5.2.1 Procedure

If the call connected event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the terminal receives an incoming call connect message (in the case of an MT call), or when the terminal sends an outgoing call connect message (in the case of an MO call), the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - call connected) command as in clause 7.5.2.2.

In the case of a call initiated through a SET UP CALL proactive command while the call connected event is part of the current event list, the terminal shall send both the TERMINAL RESPONSE related to the proactive command, and the EVENT DOWNLOAD command, in the order TERMINAL RESPONSE first, ENVELOPE (EVENT DOWNLOAD - call connected) second.

NOTE: The technology specific toolkit specification will define the appropriate call connect message.

### 7.5.2.2 Structure of ENVELOPE (EVENT DOWNLOAD - call connected)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Transaction identifier	8.28	M	Y	C

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- call connected.

Device identities:

- in the case of connecting at the near end (an MT call), the terminal shall set the device identities to:
  - source: terminal;
  - destination: UICC.
- in the case of connecting at the far end (an MO call), the terminal shall set the device identities to:
  - source: network;
  - destination: UICC.

Transaction identifier: the transaction identifier data object shall contain one transaction identifier, and this shall be the Transaction Identifier in the call connect message.

Response parameters/data:

- none.

## 7.5.3 Call disconnected event

### 7.5.3.1 Procedure

If the call disconnected event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), while the terminal is not in the NULL state (i.e. has sent or received a call setup message) and the call is disconnected, the terminal shall inform the UICC that this has occurred by using the ENVELOPE (EVENT DOWNLOAD - call disconnected) command as defined in clause 7.5.3.2. This can happen as the result of the terminal sending or receiving one or more disconnect messages, or as the result of a radio link failure; if more than one of these occur within the same call, the ENVELOPE command shall be sent on the first occurrence.

If the terminal initiates the disconnection, or in the case of radio link failure, this is considered a "near end" disconnection, whereas a "far end" disconnection is defined as when the network initiates the disconnection. The terminal shall set the Device Identities accordingly.

NOTE: The technology specific toolkit specification will define the appropriate disconnect messages.

### 7.5.3.2 Structure of ENVELOPE (EVENT DOWNLOAD - call disconnected)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C+D)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Transaction identifier	8.28	M	Y	C
Cause	8.26	O	N	D

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- call disconnected.

Device identities:

- in the case of "near end" disconnection, the terminal shall set the device identities to:
  - source: terminal;
  - destination: UICC.
- in the case of "far end" disconnection, the terminal shall set the device identities to:
  - source: network;
  - destination: UICC.

Transaction identifier: the transaction identifier data object shall contain a list of the transaction identifiers for each of the calls being disconnected.

Cause: the cause shall reflect the cause information element sent or received in the disconnect message triggering the ENVELOPE command. If the cause information element was not present in the message, or the cause data object shall not be included. In the case of a radio link timeout, the cause data object shall be included, with a value part of zero length.

Response parameters/data:

- none.

## 7.5.4 Location status event

### 7.5.4.1 Procedure

If the location status event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the terminal enters the idle state with the result that either the Location status or Location information has been changed or updated, the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - location status) command as defined in clause 7.5.4.2.

NOTE: The technology specific toolkit specification will define the appropriate idle state for change of location purposes.

### 7.5.4.2 Structure of ENVELOPE (EVENT DOWNLOAD - Location status)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C+D)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Location status	8.27	M	Y	C
Location information	8.19	C	N	D

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- location status.

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Location status: this object shall contain the current service state of the terminal.

Location information: this object shall only be included if the Location status object indicates Normal Service. This object shall contain the details of the network, location area and cell that have been selected.

Response parameters/data:

- none.

## 7.5.5 User activity event

### 7.5.5.1 Procedure

If the user activity event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then the terminal shall follow the procedure below:

- when the terminal next detects some user activity (e.g. a key-press, removal of key-lock), the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - user activity) command as defined in clause 7.5.5.2;
- as a result of sending this command to the UICC, the terminal shall remove the user activity event from its current event list. This is in order for the terminal to report this event only once after the event has been requested by the UICC.

### 7.5.5.2 Structure of ENVELOPE (EVENT DOWNLOAD - User activity)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- user activity.

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Response parameters/data:

- none.

## 7.5.6 Idle screen available event

### 7.5.6.1 Procedure

If the idle screen available event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then the terminal shall follow the procedure below:

- when the terminal next enters a state where it would accept rather than reject a DISPLAY TEXT command of normal priority, the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - idle screen available) command as defined in clause 7.5.6.2;
- as a result of sending this command to the UICC, the terminal shall remove the idle screen available event from its current event list. This is in order for the terminal to report this event only once after the event has been requested by the UICC.

### 7.5.6.2 Structure of ENVELOPE (EVENT DOWNLOAD - Idle screen available)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- idle screen available.

Device identities: the terminal shall set the device identities to:

- source: display;
- destination: UICC.

Response parameters/data:

- none.

### 7.5.7 Card reader status event

The following clauses under clause 7.5.7 apply if class "a" is supported.

#### 7.5.7.1 Procedure

If the card reader status event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the terminal detects one of the following changes:

- a card reader becomes available or unavailable (e.g. a removable card reader is attached); or
- a card is inserted or removed.

The terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - card reader status) command as defined in clause 7.5.7.2.

#### 7.5.7.2 Structure of ENVELOPE (EVENT DOWNLOAD - Card reader status)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Card reader status	8.33	M	Y	C

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- card reader status.

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Card reader status: the card reader status data object shall contain the identifier and status flags for the card reader that has generated the event.

Response parameters/data: None for this type of ENVELOPE command.

## 7.5.8 Language selection event

### 7.5.8.1 Procedure

If the language selection event is part of the event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the terminal changes the currently used language, the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - language selection) command as defined in clause 7.5.8.2.

### 7.5.8.2 Structure of ENVELOPE (EVENT DOWNLOAD - Language selection)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Language	8.45	M	Y	C

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- language selection.

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Language: This object shall contain the currently used language of the terminal.

Response parameters/data: None for this type of ENVELOPE command.

## 7.5.9 Browser termination event

The following clauses apply if class "c" is supported.

### 7.5.9.1 Procedure

If the browser termination event is part of the event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the browser is terminated either by the user action or by an error, the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - browser termination) command as defined in clause 7.5.9.2.

### 7.5.9.2 Structure of ENVELOPE (EVENT DOWNLOAD - Browser termination)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Browser termination cause	8.51	M	Y	C

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- browser termination.

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Browser termination cause: This object shall contain the browser termination cause.

Response parameters/data: None for this type of ENVELOPE command.

## 7.5.10 Data available event

The following clauses apply if class "e" is supported.

### 7.5.10.1 Procedure

If the Data available event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then, only if the targeted channel buffer is empty when new data arrives in it, the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - Data available) command as defined in clause 7.5.10.2.



### 7.5.10.2 Structure of ENVELOPE (EVENT DOWNLOAD - Data available)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C+D)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Channel status	8.56	M	Y	C
Channel data length	8.54	M	Y	D

Event list: the Event list data object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- data available.

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Channel status: this data object shall contain the status and identifier of the channel on which the event occurred.

Channel data length: this data object shall contain the number of bytes received, e.g. available in the channel buffer. If more than 255 bytes are available, "FF" is used.

Response parameters/data: None for this type of ENVELOPE command.

## 7.5.11 Channel status event

The following clauses apply if class "e" is supported.

### 7.5.11.1 Procedure

If the Channel status event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then, when the terminal detects one of the following changes:

- a TCP connection or a direct communication channel is closed for Terminal Server Mode;
- a state change in a TCP connection for UICC Server Mode (i.e. a transition to any of these states: TCP in LISTEN state, TCP in CLOSED state, TCP in ESTABLISHED state);
- a link enters an error condition;
- the user cancels the ongoing session; or
- any other error;

which is not resulting from the execution of a proactive command; or

- the link was established or link establishing Failed.

after an OPEN CHANNEL in background mode, the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - Channel status) command as defined in clause 7.5.11.2.

The channel identifier for a data channel shall not be released during a card session until the CLOSE CHANNEL command for this channel identifier has been successfully executed.

The terminal shall not empty the Rx/Tx buffers during the card session until the CLOSE CHANNEL command has been successfully executed, excepted for the UICC Server mode or terminal server mode when a TCP disconnect occurs.

### 7.5.11.2 Structure of ENVELOPE (EVENT DOWNLOAD - Channel status)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C+D+E)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Channel status	8.56	M	Y	C
Bearer Description	8.52	C	Y	D
Other address (local address)	8.58	C	Y	E

Event list: the Event list data object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- channel status.

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Channel status: this data object shall contain the status and identifier of the channel on which the event occurred.

Bearer Description: this data object shall only be present after an OPEN CHANNEL in background mode.

Other address (local address): this data object shall only be present after an OPEN CHANNEL in background mode with dynamic local address request.

Response parameters/data: None for this type of ENVELOPE command.

## 7.5.12 Access Technology Change Event

### 7.5.12.1 Procedure

If the Access Technology Change event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then, when the terminal detects a change in its current access technology or any of its access technologies the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - Access Technology Change) command as defined in clause 7.5.12.2.

If the event is set up with support for multiple access technologies, the UICC shall be informed if any of the access technologies changes. If the event is set up with support for single access technologies and with support for multiple access technologies, both Access Technology Change Event notifications shall be sent to the UICC independently.

### 7.5.12.2 Structure of ENVELOPE (EVENT DOWNLOAD - Access Technology Change)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Access Technology	8.61	M	Y	C

Event list: the Event list data object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- Access Technology Change (single access technology) or Access Technology Change (multiple access technologies).

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Access Technology: this data object shall contain the current access technology or technologies that the terminal is using.

- If a single access technology is requested in the Event Access Technology Change, the length shall be set to 1. If the terminal is currently connected via more than one technology, it shall select the most appropriate value.
- If multiple access technologies are requested in the Event Access Technology Change, the terminal shall include all access technologies it is currently connected to in the data object.

Response parameters/data: None for this type of ENVELOPE command.

## 7.5.13 Display parameters changed event

### 7.5.13.1 Procedure

If the display parameters changed event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the screen of the terminal is resized, the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - Display parameters changed) command as defined in clause 7.5.13.2.

### 7.5.13.2 Structure of ENVELOPE (EVENT DOWNLOAD - Display parameters changed)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Display Parameters	8.62	M	Y	C

Event list: the Event list data object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- display parameters changed.

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Display parameters changed: this data object shall contain the current terminal's screen parameters.

Response parameters/data: None for this type of ENVELOPE command.

## 7.5.14 Local Connection event

The following clauses apply if class "f" is supported.

### 7.5.14.1 Procedure

If the Local Connection event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the terminal receives an incoming connection request on a local bearer using a service previously declared by the UICC, the terminal shall inform the UICC that it has occurred, by using the ENVELOPE (EVENT DOWNLOAD - Local Connection) command as defined in clause 7.5.14.2. The terminal shall then wait for an OPEN CHANNEL with the parameters given in the event before proceeding with the local connection establishment.

### 7.5.14.2 Structure of ENVELOPE (EVENT DOWNLOAD - Local Connection)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1].

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C+D+E+F)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Service Record	8.63	M	Y	D
Remote Entity Address	8.68	O	N	C
UICC/terminal interface transport level	8.59	O	N	E
Remote Entity Transport Level Address	8.58	C	N	F

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- local connection.

Device identities: the terminal shall set the device identities to:

- source: Network;
- destination: UICC.

Service Record: this data object shall contain the service record of the service being connected by a remote device. If the terminal provides a Service Record different from '00', the UICC shall ignore it.

Remote Entity Address: this data object shall return the remote entity address of the remote device that is trying to connect itself.

UICC/terminal interface transport level: this data object shall contain the incoming connection transport layer protocol and the set of parameters required to make the transport connection. The data that will be received/sent from the CAT to the transport layer is a SDU that will be received/transmitted in the Transport-PDU.

Remote Entity Transport Level Address: this data object shall contain the remote entity network address (e.g. IP address). This data destination address shall be included when the UICC/terminal interface transport level is present, otherwise it is ignored.

Response parameters/data: none.

## 7.5.15 Network Search Mode Change Event

### 7.5.15.1 Procedure

If the Network Search Mode Change event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then, when the terminal detects a change in its current Network Search Mode the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - Network Search Mode Change) command as defined in clause 7.5.15.2.

### 7.5.15.2 Structure of ENVELOPE (EVENT DOWNLOAD - Network Search Mode Change)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Network search mode	8.75	M	Y	D

Event list: the Event list data object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- Network Search Mode Change.

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Network search mode: this data object shall contain the current network search mode of the mobile.

Response parameters/data: None for this type of ENVELOPE command.

## 7.5.16 Browsing status event

The following clauses apply if class "c" is supported.

### 7.5.16.1 Procedure

If the browsing status event is part of the event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the browser receives a distant error from the network, the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - browsing status) command as defined in clause 7.5.16.2.

### 7.5.16.2 Structure of ENVELOPE (EVENT DOWNLOAD - Browsing status)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Browsing status	8.77	M	Y	C

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- browsing status.

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Browsing status: This object shall contain the error status received by the browser.

Response parameters/data: None for this type of ENVELOPE command.

## 7.5.17 Frames Information changed event

The following clauses apply if class "i" is supported.

### 7.5.17.1 Procedure

If the frames information changed event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the frames are resized, the terminal shall inform the UICC that this has occurred, by using the ENVELOPE (EVENT DOWNLOAD - Frames information changed) command as defined below.

### 7.5.17.2 Structure of ENVELOPE (EVENT DOWNLOAD - Frames Information changed)

Direction: terminal to UICC.

The command header is specified in TS 131 110 [7].

Command parameters/data.

Description	Clause	M/O	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Frames Information	8.79	M	Y	C

- Event list: the Event list data object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:
  - Frames information changed.
- Device identities: the terminal shall set the device identities to:
  - source: terminal;
  - destination: UICC.
- Frames Information: this data object shall contain information regarding the terminal's current frames.

Response parameters/data: None for this type of ENVELOPE command.

## 7.5.18 HCI connectivity event

The following clauses apply if class "m" is supported.

### 7.5.18.1 Procedure

If the Contactless HCI event is part of the event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then the terminal shall follow the procedure below:

- when the terminal detects an HCI connectivity event as defined in TS 102 622 [40], it shall inform the UICC of that state by using the ENVELOPE (EVENT DOWNLOAD - HCI connectivity event) command as defined in clause 7.5.18.2.

### 7.5.18.2 Structure of ENVELOPE (EVENT DOWNLOAD - HCI connectivity event)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform and in TS 151 011 [8] for a 2G platform.

Command parameters/data.

Description	Clause	M/O	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B

Event list: the event list object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:

- HCI connectivity event.

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Response parameters/data:

- none.

## 7.5.19 Contactless state request

The following clause applies if class "r" is supported.

### 7.5.19.1 Procedure

If the contactless state request is part of the event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then the terminal shall follow the procedure below:

- The terminal shall inform the UICC that the user has issued a request to disable or enable the contactless functionality in the UICC (e.g. by pressing a dedicated (soft) button) by using the ENVELOPE (EVENT DOWNLOAD - Contactless state request) command as defined in clause 7.5.19.2.

The new state of the contactless functionality in the UICC shall be signalled using the proactive command specified in clause 6.4.41.

### 7.5.19.2 Structure of ENVELOPE(EVENT DOWNLOAD - Contactless state request)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1].

Command parameters/data.

Description	Clause	M/O	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Contactless state request	8.91	M	Y	C

- Event list: the Event list data object shall contain only one event (value part of length 1 byte), and terminal shall set the event to:
  - Contactless state request.
- Device identities: the terminal shall set the device identities to:
  - source: terminal;
  - destination: UICC.
- Contactless state request: shall contain the requested state of the contactless functionality in the UICC.

## 7.5.20 Profile Container

The following clauses apply if class "u" is supported.

Encrypted and MAC protected data structures apply only if class "x" is supported in addition.



### 7.5.20.1 Procedure

If the profile container event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the terminal contains eCAT clients that want to use eCAT to communicate with the UICC, the eCAT client shall prepare its encapsulated profile and the terminal shall inform the UICC about the facilities of each eCAT client by using the ENVELOPE (EVENT DOWNLOAD - Profile Container) command as defined in clause 7.5.20.2.

If the secured profile container event is part of the current event list (as set up by the last SET UP EVENT LIST command, see clause 6.4.16), then when the terminal contains eCAT clients that want to use an eCAT secure channel to communicate with the UICC, the terminal shall inform the UICC about this request of the eCAT client by using the ENVELOPE (EVENT DOWNLOAD - Profile Container) command as defined in clause 7.5.20.2. Once the eCAT secure channel is established, the eCAT client shall send a secured Encapsulated Profile as defined in TS 102 484 [48].

If the UICC wants to communicate with the eCAT client, it shall assign an eCAT client identity in the response. This starts an encapsulated command session with the eCAT client. Else the eCAT client identity TLV shall be empty, i.e. its length shall be zero.

Both the UICC and the eCAT client can end an encapsulated command session. Unless the end of a card session occurs, which terminates an eCAT secure channel and ends an encapsulated command session at the same time, an eCAT secure channel shall be terminated before an encapsulated command session is ended.

To avoid ambiguous situations, all encapsulated command sessions to eCAT clients shall be ended before removal of the profile container event from the current event list.

### 7.5.20.2 Structure of ENVELOPE (EVENT DOWNLOAD - Profile Container)

Direction: eCAT client to UICC.

The command header is specified in TS 102 221 [1].

Command parameters/data for a non secured Encapsulated Profile:

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C+D)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Text string (eCAT client name)	8.15	M	Y	C
eCAT client profile	8.94	M	Y	D

The event in the event list data object shall be set to Profile Container.

Command parameters/data for an Encapsulated Profile to request an eCAT secure channel:

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Text string (eCAT client name)	8.15	M	Y	C

The event in the event list data object shall be set to Secured Profile Container.

Command parameters/data for a non-encrypted secured Encapsulated Profile:

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C+D+E+F)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
eCAT sequence number	8.98	M	N	C
Text string (eCAT client name)	8.15	M	Y	D
eCAT client profile	8.94	M	Y	E
MAC	8.100	M	N	F

The event in the event list data object shall be set to Secured Profile Container.

Command parameters/data for an encrypted Encapsulated Profile:

Description	Clause	M/O/C	Min	Length
Event download tag	9.1	M	Y	1
Length (A+B+C+D)	-	M	Y	1 or 2
Event list	8.25	M	Y	A
Device identities	8.7	M	Y	B
Encrypted TLV list	8.99	M	Y	C
MAC	8.100	M	N	D

The event in the event list data object shall be set to Secured Profile Container.

Device identities: the terminal shall set the device identities to:

- source: terminal - if an encapsulated command session has not yet been started with this eCAT client; eCAT client x - if an encapsulated command session is already ongoing with this eCAT client;
- destination: UICC.

Response parameters/data for a non-encrypted Encapsulated Profile:

Description	Clause	M/O/C	Min	Length
eCAT sequence number	8.98	C	N	A
eCAT client identity	8.95	M	Y	B
CAT service list	8.102	C	N	C
MAC	8.100	C	N	D

eCAT sequence number and MAC shall only be present if the encapsulated command session uses integrity protection only.

The eCAT client identity shall contain the identifier assigned by the UICC, which is used in the device identities data object in subsequent communication with the eCAT client.

The CAT service list contains services available in the UICC for the eCAT client.

If support for the CAT service list for eCAT client is not indicated in the eCAT client profile of the command data, no CAT service list data object shall be provided by the UICC.

If support for the CAT service list for eCAT client is indicated in the eCAT client profile of the command data and the UICC supports sending the CAT service list in the response, the CAT service list data object shall be provided by the UICC.

A UICC and an eCAT client that support secured eCAT sessions shall support the CAT service list. Otherwise support is optional.

Response parameters/data for an encrypted Encapsulated Profile:

Description	Clause	M/O/C	Min	Length
Encrypted TLV list	8.99	M	Y	A
MAC	8.100	M	N	B

## 7.5.21 Void

## 7.6 MMS Transfer Status

### 7.6.1 Procedure

The terminal shall follow the procedure below (if class "j" is supported):

- when the terminal is asked by the UICC to submit a multimedia message, and after the message has been submitted by the terminal to the network, the terminal receives a "MM1\_submit.RES" message (see TS 123 140 [37]) from the network. Then the terminal shall send this "MM1\_submit.RES" message to the UICC using the ENVELOPE (MMS Transfer Status) immediately upon its reception;
- when the terminal is asked by the UICC to retrieve a multimedia message, then the terminal shall store the received multimedia message in the UICC upon its reception. Upon the completion of the storage, the terminal shall notify it to the UICC using the ENVELOPE (MMS Transfer Status). The terminal shall neither display the message nor alert the user;
- if the UICC responds with '93 00', the terminal shall consider that the ENVELOPE (MMS Transfer Status) has not been successfully transferred to the UICC. The terminal may retry the same command.

### 7.6.2 Structure of ENVELOPE (MMS Transfer Status)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
MMS transfer status tag	9.1	M	Y	1
Length (A+B+C+D+E+F)	-	M	Y	1
Device identities	8.7	M	Y	A
MMS Transfer File	8.18	M	Y	B
Multimedia Message Identifier	8.83	C	N	C
Multimedia Message Transfer Status	8.84	C	N	D
Multimedia Message Notification	8.78	C	N	E
Last Envelope	8.79	C	N	F

Device identities: the terminal shall set the device identities to:

- source: network;
- destination: UICC.

MMS Transfer File: is the path of the MMS Reception File or the MMS Submission File.

Multimedia Message Identifier: is the identifier of the Multimedia Message within the MMS Transfer File. This Identifier is mandatory in case the MMS Transfer File is able to store several MMs.

Multimedia Message Transfer Status: this data object:

- Shall contain the status of the submission of a Multimedia Message. It consists of the "MM1\_submit.RES" message described in TS 123 140 [37];

- Shall not be present in the case of a retrieval.

Multimedia Message Notification: this data object:

- Shall contain the status of the retrieval of a Multimedia Message. It consists of the "MM1\_notification.REQ" message described in TS 123 140 [37].
- Shall not be present in the case of a submission.

NOTE: The UICC is able to identify if the envelope corresponds to a previous submit or retrieve MMS by using the MMS Transfer File and the Multimedia Message Identifier that shall be the same between both commands.

Last Envelope: Indicates the last envelope sent to transmit either Multimedia Message Transfer Status or the Multimedia Message notification to the card. If one envelope is not enough to transmit all the information (i.e. the MMS notification is more than 243 bytes), the information shall be split into several ENVELOPE (MMS notification download). The final envelope is indicated by containing a Last Envelope TLV. Intermediate envelopes shall not contain this TLV.

If one envelope is enough to transmit the information, this envelope shall contain a Last Envelope TLV.

Response parameters/data: none for this type of ENVELOPE command.

## 7.7 MMS notification download

Addressing mechanism to the UICC is based on application addressing mechanism defined in TS 123 140 [37].

The application identifier to be used to target the UICC is defined by the different access technologies.

### 7.7.1 Procedure

If the service "Multimedia Messages Storage" is allocated and activated in the Service Table provided by the NAA, then the terminal shall follow the procedure below (if class "j" is supported):

When the terminal receives an MMS notification message intended to the UICC (i.e. using the application identifier defined by each access technology) then:

- the terminal shall pass the "MM1\_notification.REQ" (see TS 123 140 [37]) message to the UICC using the ENVELOPE (MMS notification download) command as defined below;
- the terminal shall wait for an acknowledgement from the UICC;
  - if the UICC responds with '90 00', terminal shall consider that the ENVELOPE (MMS notification download) has been successfully transferred to the UICC;
  - if the UICC responds with '93 00', the terminal shall consider that the ENVELOPE (MMS notification download) has not been successfully transferred to the UICC. The terminal may retry the same command;
  - if the UICC responds with '6F XX', the terminal shall consider that the ENVELOPE (MMS notification download) has not been successfully transferred to the UICC. The terminal shall not retry the same command.

If the service "Multimedia Messages Storage" is not available in the Service Table provided by the NAA, and the terminal receives an MMS Notification Message to be forwarded to the UICC, then the terminal should send an error message to the network.

If one envelope is not enough to transmit all the information (i.e. the MMS notification is more than 243 bytes), the information shall be split into several ENVELOPE (MMS notification download). The final envelope is indicated by containing a Last Envelope TLV. Intermediate envelope shall not contain this TLV.

If one envelope is enough to transmit the information, this envelope shall contain a Last Envelope TLV.

## 7.7.2 Structure of ENVELOPE (MMS notification download)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1] for a 3G platform.

Command parameters/data.

Description	Clause	M/O/C	Min	Length
MMS notification download tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Device identities	8.7	M	Y	A
Multimedia Message Notification	8.86	M	Y	B
Last Envelope	8.87	C	N	C

- Device identities: the terminal shall set the device identities to:
  - source: network;
  - destination: UICC.
- Multimedia Message Notification: The "MM1\_notification.REQ" message as specified in TS 123 140 [37];
- Last Envelope: Indicates the last envelope sent to transmit the MMS notification to the card. The presence or not of this Last Envelope TLV is described in the above procedure description of the MMS Notification download.

## 7.8 Terminal Applications

The following clauses apply if class "k" is supported.

### 7.8.1 Description

If the service "Terminal Applications" is available in the Service Table provided by the NAA, then the terminal shall follow the procedure below:

- the terminal shall inform the card of the applications present in the handset that can be granted the right to be started upon a request of the card or that are accepting direct communication channels from the UICC, by sending one or several ENVELOPE (TERMINAL APPLICATIONS) to the UICC, after each start of card session and as soon as possible when any such launch-able application is added to or removed from the terminal, or de-registered dynamically from the registry;
- if one envelope is not enough to transmit all the information (i.e. length is more than 243 bytes), the information shall be split into several ENVELOPE (TERMINAL APPLICATIONS). The final envelope is indicated by containing a Last Envelope. Intermediate envelope shall not contain this TLV. Each envelope shall be structured as defined below, i.e. it shall only contain complete comprehension TLVs and especially all mandatory ones. The chain of envelopes shall not be interrupted by another ENVELOPE command;
- if one envelope is enough to transmit the information, this envelope shall contain a Last envelope TLV;
- any set of ENVELOPE (TERMINAL APPLICATIONS) sent to the UICC replaces any previous information already received by the UICC. A set of ENVELOPE (TERMINAL APPLICATIONS) is a single sent envelope with a Last envelope tag, or a succession of envelopes with the final one containing a Last envelope tag;
- if the service "Extended Terminal Applications" is also available in the Service Table provided by the NAA, then the terminal shall only use Extended Registry Application Data objects in the ENVELOPE (TERMINAL APPLICATIONS) command(s). Else the terminal shall only use Registry Application Data objects in the ENVELOPE (TERMINAL APPLICATIONS) command(s). If the service "Extended Terminal Applications" is available in the Service Table provided by the NAA, the service "Terminal Applications" shall also be available.

NOTE: Application providers willing to have their application considered as launch-able or not by the terminal should have a way to inform the terminal about it. However, it is not in the scope of SCP to define how such a mechanism should occur. If no indication is given to the terminal, it is up to the terminal to decide whether or not to declare an application to the card as launch-able.

## 7.8.2 Structure of ENVELOPE (TERMINAL APPLICATIONS)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1].

Command parameters/data.

Description	Clause	M/O/C	Min	Length
Terminal application tag	9.1	M	Y	1
Length (A+B1+ ... + Bn+C)	-	M	Y	1 or 2
Device identities	8.7	M	Y	A
Registry application data 1 or extended registry application data 1	8.88 or 8.93	O	N	B1
...				
Registry application data n or extended registry application data n	8.88 or 8.93	O	N	Bn
Last envelope tag	8.87	C	N	C

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

Registry application data: provides information about each application on the terminal that can be launched by the UICC.

Last envelope tag: indicates that the last information concerning terminal applications has been sent.

An empty ENVELOPE (TERMINAL APPLICATIONS) (i.e. without any Registry data) is an indication for the UICC that all launch-able applications are removed or disabled in the terminal.

Response parameters/data:

- none.

## 7.9 Envelope Container

The following clauses apply if class "u" is supported.

Encrypted and MAC protected data structures apply only if class "x" is supported in addition.

### 7.9.1 Description

When an eCAT client in the terminal wants to send an ENVELOPE to the UICC containing an event or another envelope that was set up for the eCAT client, the terminal shall inform the UICC about the ENVELOPE of the eCAT client, by encapsulating it in the ENVELOPE (ENVELOPE CONTAINER) command as defined in clause 7.9.2.

Services activated in the Service Table provided by the NAA do not apply to envelopes transmitted in an ENVELOPE (ENVELOPE CONTAINER). If supported by eCAT client and UICC, services are indicated in the CAT service list for the eCAT client. Else it is specific to an eCAT client which encapsulated envelopes it will send to the UICC.

If an eCAT client wants to end a non-secured encapsulated command session, it shall send the ENVELOPE (ENVELOPE CONTAINER) command with an encapsulated envelope type of zero length.

If an eCAT client wants to terminate an eCAT secure channel, it shall send a secured ENVELOPE (ENVELOPE CONTAINER) command with an encapsulated envelope type of zero length. This shall terminate the an eCAT secure channel, but not end the encapsulated command session. The latter can be ended by the UICC or the eCAT client with a separate command.

## 7.9.2 Structure of ENVELOPE (ENVELOPE CONTAINER)

Direction: eCAT client to UICC.

The command header is specified in TS 102 221 [1].

Command parameters/data for a non-encrypted Envelope Container:

Description	Clause	M/O/C	Min	Length
Envelope container application tag	9.1	M	Y	1
Length (A+B+C++D+E <sub>1</sub> ... E <sub>N</sub> +F)	-	M	Y	1 or 2
Device identities	8.7	M	Y	A
eCAT sequence number	8.98	C	N	B
Encapsulated envelope type	8.96	M	Y	C
Device identities of encapsulated envelope	8.7	M	Y	D
Other comprehension TLVs of encapsulated envelope	...	...	...	E <sub>1</sub> ... E <sub>N</sub>
MAC	8.100	C	N	F

To avoid nested structures for comprehension TLVs, the BER TLV length is not provided for the encapsulated envelope (it can be derived from the overall length) and the tag is provided in the separate comprehension TLV Encapsulated Envelope Type.

eCAT sequence number and MAC shall only be present if the encapsulated command session uses integrity protection only.

Command parameters/data for an encrypted Envelope Container.

Description	Clause	M/O/C	Min	Length
Envelope container application tag	9.1	M	Y	1
Length (A+B+C)	-	M	Y	1 or 2
Device identities	8.7	M	Y	A
Encrypted TLV list	8.99	M	Y	B
MAC	8.100	M	N	C

Device identities: the terminal shall set the device identities to:

- source: eCAT client;
- destination: UICC.

Response parameters/data in case of no security:

If response data of the encapsulated envelope is empty, the response data of the Envelope Container is also empty.

Else the response data of the Envelope Container shall be identical to the list of comprehension TLV data objects defined for the response to a non-encapsulated envelope. If the response for a non-encapsulated envelope is not in that format, a new response format shall be defined for the encapsulated command.

Response parameters/data in case of integrity protection only:

Description	Clause	M/O/C	Min	Length
eCAT sequence number	8.98	M	N	A
List of comprehension TLVs as specified for non-secured Envelope Container (possibly empty)	...	...	...	B <sub>1</sub> ... B <sub>N</sub>
MAC	8.100	M	N	C

Response parameters/data in case of encryption:

Description	Clause	M/O/C	Min	Length
Encrypted TLV list	8.99	M	Y	A
MAC	8.100	M	N	B

## 7.10 Service List Retrieval

The following clauses apply if class "y" is supported.

### 7.10.1 Description

In the context of a CAT APDU secure channel, this envelope provides a secure method for the terminal to retrieve CAT related information stored in the service tables of NAAs, which otherwise can only be read in an unsecured manner.

This envelope shall be supported by UICCs implementing the Platform to Platform CAT APDU secure channel. The terminal shall only send such an envelope inside a CAT APDU secure channel session.

If any service in one of the NAA's service tables which the terminal has retrieved with this envelope changes (e.g. by an RFM action), the UICC shall issue a REFRESH command with a setting that requests a restart of the CAT session. This ensures that the data in the service tables and the data retrieved via the CAT service lists are consistent.

### 7.10.2 Structure of ENVELOPE (SERVICE LIST)

Direction: terminal to UICC.

The command header is specified in TS 102 221 [1].

Command parameters/data:

Description	Clause	M/O/C	Min	Length
Service list envelope tag	9.1	M	Y	1
Length (A+B)	-	M	Y	1 or 2
Device identities	8.7	M	Y	A
NAA AID	8.60	M	Y	B

Device identities: the terminal shall set the device identities to:

- source: terminal;
- destination: UICC.

NAA AID: the AID of the NAA for which the terminal wants to retrieve the CAT service list.

Response parameters/data:

Description	Clause	M/O/C	Min	Length
CAT Service List	8.102	M	Y	A

CAT Service List: this data object shall contain the services related to the NAA indicated in the NAA AID.



## 8 COMPREHENSION-TLV data objects

This clause specifies the coding of the COMPREHENSION-TLV data objects, which are contained in a BER-TLV data object. COMPREHENSION-TLV data objects may be transferred across the interface in either direction. A COMPREHENSION-TLV data object consists of a tag of length one byte, a length indicator, which gives the number of bytes in the value field, and a value part of variable length, whose contents, meaning and coding are given below.

Tag codings are given in clause 9.3 for all COMPREHENSION-TLV data objects.

'00' and 'FF' are never used as tag values for COMPREHENSION-TLVs. This is in alignment with TS 101 220 [31]. Padding characters are not allowed.

For some of the COMPREHENSION-TLV data objects described, the length field shall be coded on 1 or 2 bytes (Y value) according to annex C, depending on the value of byte 1.

All bits and bytes indicated as RFU within all COMPREHENSION-TLV data objects shall be respectively set to 0 and '00' by the sending entity.

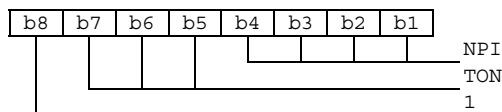
The handling of reserved values and RFU bits or bytes within all COMPREHENSION-TLV data objects at the receiving entity is described in clause 6.10.

### 8.1 Address

Byte(s)	Description	Length
1	Address tag	1
2 to (Y-1)+2	Length (X)	Y
(Y-1)+3	TON and NPI	1
(Y-1)+4 to (Y-1)+X+2	Dialling number string	X-1

TON/NPI:

- Contents:
  - Type of Number (TON) and numbering plan identification (NPI).
- Coding:
  - It is coded as defined for  $EF_{ADN}$  in TS 131 102 [6], the format of the byte is as follows:



TON values:

- Coding:
  - 000: Unknown;
  - 001: International Number;
  - 010: National Number;
  - 011: Network Specific Number;
  - other values are reserved or access technology specific.

NPI values:

- Coding:
  - 0000: Unknown;
  - 0001: ISDN/telephony numbering plan (Recommendation ITU-Ts E.164 [22] and E.163 [i.3]);
  - 0011: Data numbering plan (Recommendation ITU-T X.121 [23]);
  - 0100: Telex numbering plan (Recommendation ITU-T F.69 [24]);
  - 1001: Private numbering plan;
  - 1111: Reserved for extension;
  - other values are reserved or access technology specific.

Dialling number string:

- Coding:
  - coded as for  $EF_{ADN}$ , except that the length is variable as given above, i.e. the length is not fixed to 10 bytes as specified for  $EF_{ADN}$  and that no extension mechanism is used. It may include DTMF separators and DTMF digits, which the terminal shall send in the same way as for  $EF_{ADN}$  but without locally generating audible DTMF tones to the user;
  - see TS 131 102 [6] for the coding of  $EF_{ADN}$ .

## 8.2 Alpha identifier

Byte(s)	Description	Length
1	Alpha identifier tag	1
2 to (Y-1)+2	Length (X)	Y1
(Y-1)+3 to (Y-1)+X+2	Alpha identifier	X

- Coding:
  - the alpha identifier is coded as for  $EF_{ADN}$ ;
  - see TS 131 102 [6] for the coding of  $EF_{ADN}$ .

## 8.3 Subaddress

Byte(s)	Description	Length
1	Subaddress tag	1
2 to (Y-1)+2	Length (X)	Y
(Y-1)+3 to (Y-1)+X+2	Subaddress	X

Subaddress contains information as for  $EF_{ADN}$ . See TS 131 102 [6] for the coding of. All information defined in TS 131 102 [6] shall be given in the value part of the data object, except the length of subaddress contents (which is given here by the length part of the data object).

## 8.4 Capability configuration parameters

Byte(s)	Description	Length
1	Capability configuration parameters tag	1
2 to (Y-1)+2	Length (X)	Y
(Y-1)+3 to (Y-1)+X+2	Capability configuration parameters	X

Capability configuration parameters coding is defined by the different access technologies.

## 8.5 Void

## 8.6 Command details

Byte(s)	Description	Length
1	Command details tag	1
2	Length = '03'	1
3	Command number	1
4	Type of command	1
5	Command Qualifier	1

Command number:

- Contents and coding: see clause 6.5.1.

Type of command:

- Contents:
  - the Type of Command specifies the required interpretation of the data objects which follow, and the required terminal procedure;
- Coding:
  - see clause 9.4;
  - the terminal shall respond to reserved values (i.e. values not listed) with the result "Command type not understood".

Command Qualifier:

- Contents: qualifiers specific to the command;
- Coding:
  - REFRESH:
    - '00' = NAA Initialization and Full File Change Notification;
    - '01' = File Change Notification;
    - '02' = NAA Initialization and File Change Notification;
    - '03' = NAA Initialization;
    - '04' = UICC Reset;
    - '05' = NAA Application Reset, only applicable for a 3G platform;

- '06' = NAA Session Reset, only applicable for a 3G platform;
- '07' = Reserved by 3GPP ("Steering of Roaming" REFRESH support);
- '08' = Reserved by 3GPP (Steering of Roaming for I-WLAN);
- '09' to 'FF' = reserved values.
- MORE TIME: this byte is RFU.
- POLL INTERVAL: this byte is RFU.
- POLLING OFF: this byte is RFU.
- SET UP CALL:
  - '00' = set up call, but only if not currently busy on another call;
  - '01' = set up call, but only if not currently busy on another call, with redial;
  - '02' = set up call, putting all other calls (if any) on hold;
  - '03' = set up call, putting all other calls (if any) on hold, with redial;
  - '04' = set up call, disconnecting all other calls (if any);
  - '05' = set up call, disconnecting all other calls (if any), with redial;
  - '06' to 'FF' = reserved values.
- SEND DTMF: this byte is RFU.
- SEND SHORT MESSAGE:
  - bit 1:       0 = packing not required;  
              1 = SMS packing by the terminal required.
  - bits 2 to 8: = 0 RFU.
- SET UP EVENT LIST: this byte is RFU.
- PLAY TONE:
  - bit 1:       0 = use of vibrate alert is up to the terminal;
  - 1 = vibrate alert, if available, with the tone.
  - bits 2 to 8: = 0 RFU.
- DISPLAY TEXT:
  - bit 1:       0 = normal priority;  
              1 = high priority.
  - bits 2 to 7: = RFU.
  - bit 8:       0 = clear message after a delay;  
              1 = wait for user to clear message.
- GET INKEY:
  - bit 1:       0 = digits (0 to 9, \*, # and +) only;  
              1 = alphabet set.

- bit 2: 0 = SMS default alphabet;  
1 = UCS2 alphabet.
  - bit 3: 0 = character sets defined by bit 1 and bit 2 are enabled;  
1 = character sets defined by bit 1 and bit 2 are disabled and the "Yes/No" response is requested.
  - bit 4: 0 = user response shall be displayed. The terminal may allow alteration and/or confirmation;  
1 = an immediate digit response (0 to 9, \* and #) is requested.
  - bits 5 to 7: = RFU.
  - bit 8: 0 = no help information available;  
1 = help information available.
- GET INPUT:
- bit 1: 0 = digits (0 to 9, \*, #, and +) only;  
1 = alphabet set.
  - bit 2: 0 = SMS default alphabet;  
1 = UCS2 alphabet.
  - bit 3: 0 = terminal may echo user input on the display;  
1 = user input shall not be revealed in any way (see note).
  - bit 4: 0 = user input to be in unpacked format;  
1 = user input to be in SMS packed format.
  - bits 5 to 7: = RFU.
  - bit 8: 0 = no help information available;  
1 = help information available.

NOTE: Where user input is not to be revealed, the terminal may provide an indication of key entries, such as by displaying "\*". See clause 6.4.3 for more information on the character set available in this mode.

- SELECT ITEM:
- bit 1: 0 = presentation type is not specified;  
1 = presentation type is specified in bit 2.
  - bit 2: 0 = presentation as a choice of data values if bit 1 = '1';  
1 = presentation as a choice of navigation options if bit 1 is '1'.
  - bit 3: 0 = no selection preference;  
1 = selection using soft key preferred.
  - bits 4 to 7: = RFU.
  - bit 8: 0 = no help information available;  
1 = help information available.

- SET UP MENU:
  - bit 1: 0 = no selection preference;  
1 = selection using soft key preferred.
  - bits 2 to 7: = RFU.
  - bit 8: 0 = no help information available;  
1 = help information available.
- PROVIDE LOCAL INFORMATION:
  - '00' = Location Information according to current NAA;
  - '01' = IMEI of the terminal;
  - '02' = Network Measurement results according to current NAA;
  - '03' = Date, time and time zone;
  - '04' = Language setting;
  - '05' = Reserved for GSM;
  - '06' = Access Technology (single access technology);
  - '07' = ESN of the terminal;
  - '08' = IMEISV of the terminal;
  - '09' = Search Mode;
  - '0A' = Charge State of the Battery (if class "g" is supported);
  - '0B' = MEID of the terminal;
  - '0C' = reserved for 3GPP (current WSID);
  - '0D' = Broadcast Network information according to current Broadcast Network Technology used;
  - '0E' = Multiple Access Technologies;
  - '0F' = Location Information for multiple access technologies;
  - '10' = Network Measurement results for multiple access technologies;
  - '11' to 'FF' = Reserved.
- SET UP IDLE MODE TEXT: this byte is RFU.
- PERFORM CARD APDU: this byte is RFU.
- POWER OFF CARD: this byte is RFU.
- POWER ON CARD: this byte is RFU.
- GET READER STATUS:
  - '00' = Card reader status;
  - '01' = Card reader identifier;
  - '02' to 'FF' = Reserved.

- TIMER MANAGEMENT:
  - bits 1 to 2: 00 = start;  
01 = deactivate;  
10 = get current value;  
11 = RFU.
  - bits 3 to 8: RFU.
- RUN AT COMMAND: this byte is RFU.
- LANGUAGE NOTIFICATION:
  - bit 1: 0 = non-specific language notification;  
1 = specific language notification.
  - bits 2 to 8: = RFU.
- LAUNCH BROWSER:
  - '00' = launch browser if not already launched;
  - '01' = not used;
  - '02' = use the existing browser (the browser shall not use the active existing secured session);
  - '03' = close the existing browser session and launch new browser session;
  - '04' = not used;
  - '05' to 'FF' = RFU.
- OPEN CHANNEL for CS, packet data service, local and Default (network) bearer:
  - bit 1: 0 = on demand link establishment;  
1 = immediate link establishment.
  - bit 2: 0 = no automatic reconnection;  
1 = automatic reconnection.
  - bit 3: 0 = no background mode;  
1 = immediate link establishment in background mode (bit 1 is ignored).
  - bits 4 to 8: = RFU.
- OPEN CHANNEL for UICC Server Mode:
  - This byte is RFU.
- OPEN CHANNEL for Terminal Server Mode:
  - bit 1: 0 = launch application immediately without additional launch parameters;  
1 = launch parameters following.
  - bits 2 to 8: = RFU.
- CLOSE CHANNEL for CS, packet data service, local and Default (network) bearer and Terminal Server Mode:
  - This byte is RFU.

- CLOSE CHANNEL: for UICC Server Mode:
- bit 1: 0 = close the TCP connection and go to "TCP in CLOSED state";  
1 = close the TCP connection and go to "TCP in LISTEN state".
  - bits 2 to 8: = RFU.
- RECEIVE DATA: this byte is RFU.
- SEND DATA:
- bit 1: 0 = store data in Tx buffer;  
1 = send data immediately.
  - bits 2 to 8: = RFU.
- GET CHANNEL STATUS: this byte is RFU.
- SERVICE SEARCH (if class "f" is supported): this byte is RFU.
- GET SERVICE INFORMATION (if class "f" is supported): this byte is RFU.
- DECLARE SERVICE (if class "f" is supported):
- bit 1: 0 = add a new service to the terminal service database;  
1 = delete a service from the terminal service database.
  - bit 2 to 8: = RFU
- SET FRAMES:
- '00' = This value tells the terminal to draw a separator between every adjoining frame;
  - '01' = This value tells the terminal not to draw a separator between every adjoining frame.
- GET FRAMES STATUS: this byte is RFU.
- DISPLAY MULTIMEDIA MESSAGE:
- bit 1: 0 = normal priority;  
1 = high priority.
  - bits 2 to 7: = RFU.
  - bit 8: 0 = clear message after a delay;  
1 = wait for user to clear message.
- ACTIVATE: this byte is RFU.
- CONTACTLESS STATE CHANGED: this byte is RFU.
- COMMAND CONTAINER: this byte is RFU.
- ENCAPSULATED SESSION CONTROL:
- '00' = end encapsulated command session;
  - '01' = request Master SA setup;
  - '02' = request Connection SA setup;
  - '03' = request Secure Channel Start;
  - '04' = close Master and Connection SA, keep encapsulated command session;



- all other values = RFU.

The terminal shall respond to reserved values with the result "Command type not understood".

## 8.7 Device identities

Byte(s)	Description	Length
1	Device identities tag	1
2	Length = '02'	1
3	Source device identity	1
4	Destination device identity	1

Source device identity:

- Contents:
  - the source device for information held in the data objects which follow.

Destination device identity:

- Contents:
  - the destination device for information held in the data objects which follow.

NOTE: Only some combinations of Type of Command, Data Download type and Device identities are allowed. These are defined in clause 10.

- Coding:
  - both Source and Destination device identities are coded as follows:
    - '01' = Keypad;
    - '02' = Display;
    - '03' = Earpiece;
    - '10' to '17' = Additional Card Reader x (0 to 7). Value assigned by terminal;
    - '21' to '27' = Channel with Channel identifier x (1 to 7). Value assigned by terminal in the Channel status Comprehension TLV of the TERMINAL RESPONSE following an OPEN CHANNEL command;
    - '31' to '3F' = eCAT client identifier (1 to F);
    - '81' = UICC;
    - '82' = terminal;
    - '83' = network;
    - All other values are reserved.

## 8.8 Duration

Byte(s)	Description	Length
1	Duration tag	1
2	Length = '02'	1
3	Time unit	1
4	Time interval	1

Time unit:

- Contents:
  - time unit used; minutes, seconds or tenths of seconds.
- Coding:
  - '00' minutes;
  - '01' seconds;
  - '02' tenths of seconds;
  - All other values are reserved.

Time interval:

- Contents:
  - the length of time required, expressed in units;
- Coding.

The time interval is coded in integer multiples of the time unit used. The range is from 1 unit to 255 units. The encoding is:

- '00': reserved;
- '01': 1 unit;
- '02': 2 units;
- 'FF': 255 units.

## 8.9 Item

Byte(s)	Description	Length
1	Item tag	1
2 to (Y-1)+2	Length (X)	Y
(Y-1)+3	Identifier of item	1
(Y-1)+4 to (Y-1)+X+2	Text string of item	X - 1

The identifier is a single byte between '01' and 'FF'. Each item shall have a unique identifier within an Item list.

The text string is coded in the same way as the alpha identifier for EF<sub>ADN</sub>. Any unused bytes at the end of the value part shall be coded 'FF'.

## 8.10 Item identifier

Byte(s)	Description	Length
1	Item identifier tag	1
2	Length = '01'	1
3	Identifier of item chosen	1

The identifier is a single byte between '01' and 'FF', exactly the same as for the Item data object. A null item identifier is coded '00'.

## 8.11 Response length

Byte(s)	Description	Length
1	Response length tag	1
2	Length = '02'	1
3	Minimum length of response	1
4	Maximum length of response	1

The range of length is between '00' and 'FF'. A minimum length coding of '00' indicates that there is no minimum length requirement; a maximum length coding of 'FF' indicates that there is no maximum length requirement. If a fixed length is required the minimum and maximum values are identical.

NOTE: It is not recommended to set the "Maximum length of response" to '00', as this will lead to unpredictable behaviour in the terminal.

## 8.12 Result

Byte(s)	Description	Length
1	Result tag	1
2 to (Y-1)+2	Length (X)	Y
(Y-1)+3	General result	1
(Y-1)+4 to (Y-1)+X+2	Additional information on result	X-1

General result:

- Contents:
  - General result specifies the result and indicates appropriate UICC action.
- Coding:
  - '00' = Command performed successfully;
  - '01' = Command performed with partial comprehension;
  - '02' = Command performed, with missing information;
  - '03' = REFRESH performed with additional EFs read;
  - '04' = Command performed successfully, but requested icon could not be displayed;
  - '05' = Command performed, but modified by call control by NAA;
  - '06' = Command performed successfully, limited service;
  - '07' = Command performed with modification;
  - '08' = REFRESH performed but indicated NAA was not active;
  - '09' = Command performed successfully, tone not played;
  - '10' = Proactive UICC session terminated by the user;
  - '11' = Backward move in the proactive UICC session requested by the user;
  - '12' = No response from user;
  - '13' = Help information required by the user;
  - '14' = reserved for GSM/3G.

Results '0X' and '1X' indicate that the command has been performed:

- '20' = terminal currently unable to process command;
- '21' = Network currently unable to process command;
- '22' = User did not accept the proactive command;
- '23' = User cleared down call before connection or network release;
- '24' = Action in contradiction with the current timer state;
- '25' = Interaction with call control by NAA, temporary problem;
- '26' = Launch browser generic error code;
- '27' = MMS temporary problem.

Results '2X' indicate to the UICC that it may be worth re-trying the command at a later opportunity:

- '30' = Command beyond terminal's capabilities;
- '31' = Command type not understood by terminal;
- '32' = Command data not understood by terminal;
- '33' = Command number not known by terminal;
- '34' = reserved for GSM/3G;
- '35' = reserved for GSM/3G;
- '36' = Error, required values are missing;
- '37' = reserved for GSM/3G;
- '38' = MultipleCard commands error;
- '39' = Interaction with call control by NAA, permanent problem;
- '3A' = Bearer Independent Protocol error;
- '3B' = Access Technology unable to process command;
- '3C' = Frames error;
- '3D' = MMS Error.

Results '3X' indicate that it is not worth the UICC re-trying with an identical command, as it will only get the same response. However, the decision to retry lies with the application.

The application should avoid a rapid sequence of repeated retried commands as this may be detrimental to terminal performance.

All other values are reserved.

Additional information.

- Contents:

- For the general result "Command performed successfully", some proactive commands require additional information in the command result. This is defined in the clauses below. For the general results '20', '21', '26', '38', '39', '3A', '3C' and '3D', it is mandatory for the terminal to provide a specific cause value as additional information, as defined in the clauses below. For the other general results, the terminal may optionally supply additional information. If additional information is not supplied, then the length of the value part of the data object need only contain the general result.

## 8.12.1 Void

## 8.12.2 Additional information for terminal problem

For the general result "terminal currently unable to process command", it is mandatory for the terminal to provide additional information, the first byte of which to be as defined below:

- '00' = No specific cause can be given;
- '01' = Screen is busy;
- '02' = terminal currently busy on call;
- '03' = reserved for GSM/3G;
- '04' = No service;
- '05' = Access control class bar;
- '06' = Radio resource not granted;
- '07' = Not in speech call;
- '08' = reserved for GSM/3G;
- '09' = terminal currently busy on SEND DTMF command;
- '0A' = No NAA active.

All other values shall be interpreted by the UICC as '00'. The coding '00' shall only be used by the terminal if no others apply.

## 8.12.3 Additional information for network problem

For the general result "network currently unable to process command", it is mandatory for the terminal to provide additional information.

The information provided and its coding is specific to the NAA. Bit 8 shall be set to '1'. One further value is defined:

- '00' = No specific cause can be given.

All other values shall be interpreted by the UICC as '00'. The coding '00' shall only be used by the terminal if no others apply.

## 8.12.4 Void

## 8.12.5 Void

## 8.12.6 Void

## 8.12.7 Void

## 8.12.8 Additional information for interaction with call control

For the general result "interaction with call control by NAA, permanent problem", it is mandatory for the terminal to provide additional information, the first byte of which to be as defined below:

- '00' = No specific cause can be given;
- '01' = Action not allowed;

- '02' = The type of request has changed.

All other values shall be interpreted by the UICC as '00'. The coding '00' shall only be used by the terminal if no others apply.

### 8.12.9 Additional information for MultipleCard commands

This clause applies if class "a" is supported.

For the general result "MultipleCard commands error", it is mandatory for the terminal to provide additional information, the first byte of which is defined below:

- '00' = No specific cause can be given;
- '01' = Card reader removed or not present;
- '02' = Card removed or not present;
- '03' = Card reader busy;
- '04' = Card powered off;
- '05' = C-APDU format error;
- '06' = Mute card;
- '07' = Transmission error;
- '08' = Protocol not supported;
- '09' = Specified reader not valid.

All other values shall be interpreted by the UICC as '00'. The coding '00' shall only be used by the terminal if no others apply.

### 8.12.10 Additional information for launch browser problem

For the general result "launch browser generic error code", it is mandatory for the terminal to provide additional information, the first byte of which to be as defined below:

- '00' = No specific cause can be given;
- '01' = Bearer unavailable;
- '02' = Browser unavailable;
- '03' = terminal unable to read the provisioning data.

All other values shall be interpreted by the UICC as '00'. The coding '00' shall only be used by the terminal if no others apply.

### 8.12.11 Additional information for Bearer Independent Protocol

This clause applies if class "e" or "f" is supported.

For the general result "Bearer Independent Protocol error", it is mandatory for the terminal to provide additional information, the first byte of which is defined below:

- '00' = No specific cause can be given;
- '01' = No channel available;
- '02' = Channel closed;

- '03' = Channel identifier not valid;
- '04' = Requested buffer size not available;
- '05' = Security error (unsuccessful authentication);
- '06' = Requested UICC/terminal interface transport level not available;
- '07' = remote device is not reachable (not present, not physically connected, switched off, etc.);
- '08' = Service error (service not available on remote device);
- '09' = Service identifier unknown;
- '10' = Port not available (applicable for OPEN CHANNEL related to UICC Server Mode) and Terminal Server Mode;
- '11' = Launch parameters missing or incorrect (applicable for OPEN CHANNEL or SEND DATA related to Terminal Server Mode);
- '12' = Application launch failed (applicable for SEND DATA related to Terminal Server Mode).

All other values shall be interpreted by the UICC as '00'. The coding '00' shall only be used by the terminal if no others apply.

### 8.12.12 Additional information for Frames commands

This clause applies only if class "i" is supported.

For the general result "Frames error", it is mandatory for the terminal to provide additional information, the first byte of which is defined below:

- '00' = No specific cause can be given;
- '01' = Frame identifier is not valid;
- '02' = Number of frames beyond the terminal's capabilities;
- '03' = No Frame defined;
- '04' = Requested size not supported;
- '05' = Default Active Frame is not valid.

All other values shall be interpreted by the UICC as '00'.

The coding '00' shall only be used by the terminal if no others apply.

### 8.12.13 Additional information for SUBMIT and RETRIEVE MULTIMEDIA MESSAGE

This clause applies if class "j" is supported.

For the general result "MMS error", it is mandatory for the terminal to provide additional information, the first byte of which is defined below:

- '00' = No specific cause can be given.

All other values shall be interpreted by the UICC as '00'. The coding '00' shall only be used by the terminal if no others apply.

## 8.13 3GPP- SMS TPDU

Contents and coding: see TS 131 111 [26].

## 8.14 Void

## 8.15 Text string

Byte(s)	Description	Length
1	Text string tag	1
2 to (Y-1)+2	Length (X)	Y
(Y-1)+3	Data coding scheme	1
(Y-1)+4 to (Y-1)+X+2	Text string	X-1

A null text string shall be coded with Length = '00', and no Value part.

Data coding scheme is coded as for SMS Data coding scheme defined in TS 123 038 [3].

The following Data coding scheme values are recommended:

- '00': GSM default alphabet 7 bits packed;
- '04': GSM default alphabet 8 bits;
- '08': UCS2.

### 8.15.1 Coding of text in unpacked format

This is indicated by the data coding scheme having a value of '04' GSM default alphabet 8 bits data.

This string use the SMS default 7-bit coded alphabet as defined in TS 123 038 [3] with bit 8 set to 0. It may or may not include formatting characters, but all such formatting characters shall be taken from the set given in the SMS alphabet.

NOTE: This is exactly the same format as is used for EF<sub>ADN</sub> alpha-identifiers. It is also the same as SMS messages that have been "unpacked".

### 8.15.2 Coding of text in packed format

This is indicated by the data coding scheme having a value of '00' SMS default alphabet 7 bits packed.

This string shall use the SMS default 7-bit coded alphabet, packed into 8-bit octets, as defined in TS 123 038 [3]. It may or may not include formatting characters, but all such formatting characters shall be taken from the set given in the SMS alphabet.

If the total number of characters in the text string equals  $(8n-1)$  where  $n = 1, 2, 3, \text{etc.}$  then there are 7 spare bits at the end of the message. To avoid the situation where the receiving entity confuses 7 binary zero pad bits as the @ character, the carriage return (i.e. <CR>) character shall be used for padding in this situation, as defined in TS 123 038 [3].

NOTE: This is the same format as is used in SMS messages to and from the network.

### 8.15.3 Coding of text in 16 bits UCS2 alphabet format

This is indicated by the data coding scheme having a value of '08' 16 bit UCS2 alphabet.

This string shall use the UCS2 alphabet if the UCS2 is supported, as defined in TS 123 038 [3]. It may or may not include formatting characters, but all such formatting characters shall be taken from the set given in the UCS2 alphabet.

NOTE: This is the same format as is used in SMS messages to and from the network.



## 8.16 Tone

Byte(s)	Description	Length
1	Tone tag	1
2	Length = '01'	1
3	Tone	1

- Tone
- Contents:
  - Tones can be either the standard supervisory tone, as defined in the appropriate technology specific standards, or proprietary tones defined by the terminal manufacturer. The code values for proprietary tones shall be supported by the terminal. If proprietary tones are not supported the terminal shall map these codings to tones that it can generate. The tones to be used are left as an implementation decision by the manufacturer.
- Coding:
  - Standard supervisory tones:
    - '01' Dial tone;
    - '02' Called subscriber busy;
    - '03' Congestion;
    - '04' Radio path acknowledge;
    - '05' Radio path not available/Call dropped;
    - '06' Error/Special information;
    - '07' Call waiting tone;
    - '08' Ringing tone.
  - Terminal proprietary tones:
    - '10' General beep;
    - '11' Positive acknowledgement tone;
    - '12' Negative acknowledgement or error tone;
    - '13' Ringing tone as selected by the user for incoming speech call;
    - '14' Alert tone as selected by the user for incoming SMS;
    - '15' Critical Alert - This tone is to be used in critical situations. The terminal shall make every effort to alert the user when this tone is indicated independent from the volume setting in the terminal;
    - '20' vibrate only, if available.
  - Themed tones:
    - '30' happy tone;
    - '31' sad tone;
    - '32' urgent action tone;
    - '33' question tone;
    - '34' message received tone.

- Melody tones:
  - '40' Melody 1;
  - '41' Melody 2;
  - '42' Melody 3;
  - '43' Melody 4;
  - '44' Melody 5;
  - '45' Melody 6;
  - '46' Melody 7;
  - '47' Melody 8.

The Melody tones are intended to allow the UICC to cause the terminal to play tunes.

The tones '30' to '47' may be configurable by the user through a terminal interface.

All other values are reserved.

## 8.17 Void

## 8.18 File list

Byte(s)	Description	Length
1	File List tag	1
2 to (Y-1)+2	Length (X) of bytes following	Y
(Y-1)+3	Number of files (n)	1
(Y-1)+4 to (Y-1)+X+2	Files	X-1

Number of files:

- Contents:
  - this is the number of files that will be described in the following list.

Files:

- full paths are given to files. Each of these shall be at least 4 octets in length (e.g. '3F002FE2' or '3F007F106F3A'). Each entry in the file description is composed of two bytes, where the first byte identifies the type of file (see TS 102 221 [1] or TS 151 011 [8]);
- the path '3F007FFF' indicates the relevant NAA Application dedicated file;
- an entry in the file description shall therefore always begin with '3FXX'. There can be any number of Dedicated File entries between the Master File and Elementary File. There shall be no delimiters between files, as this is implied by the fact that the full path to any EF starts with '3FXX' and ends with an Elementary type file.

## 8.19 Location information

Byte(s)	Description	Length
1	Location Information tag	1
2	Length = X	1
3 to 2+X	Access technology specific Location Information	X

Location information coding is defined by the different access technologies.

## 8.20 IMEI

Byte(s)	Description	Length
1	IMEI tag	1
2	Length = '08'	1
3 to 10	IMEI of the terminal	8

The IMEI is coded in the same manner as the value part of the Mobile Identity information element as specified in TS 124 008 [20]. The IMEI itself is specified in TS 123 003 [32].

## 8.21 Help request

Byte(s)	Description	Length
1	Help Request tag	1
2	Length = '00'	1

## 8.22 Network measurement results

Byte(s)	Description	Length
1	Network Measurement Results tag	1
2	Length = '10'	1
3 to 18	Network Measurement Results	16

- Coding:
  - defined by the different access technologies.

## 8.23 Default text

The coding of this data object is the same as for the Text String data object (see clause 8.15) with the exception that the Default Text tag has a specific value (see clause 9.3).

## 8.24 Items next action indicator

Byte(s)	Description	Length
1	Items Next Action Indicator tag	1
2	Length (X)	1
3 to 3+X-1	Items Next Action Indicator list	X

- Contents:
  - Each item of a list of items has a next action indicator coded on one byte. The length of the Items Next Action Indicator list shall be the number of items of the list of items (X shall be the number of items in the list). The order of each item next action indicator shall reflect the order of the items in the list of items.
  - The Item Next action indicator gives the possible actions that will be initiated by the UICC in case of selection by the user.
- Coding:
  - If the value is equal to '00' or if the value is reserved (that is, value not listed), the terminal shall ignore the next action indicator type.

See clause 9.4 for further information.

EXAMPLE: For the following list of items:

- item #1;
- item #2;
- item #3;
- ....;
- item #n.

The Items Next Action Indicator (NAI) shall be as follows:

Tag	Length	NAI#1	NAI#2	NAI#3	...	NAI#n
-----	--------	-------	-------	-------	-----	-------

## 8.25 Event list

Byte(s)	Description	Length
1	Event list tag	1
2 to Y+1	Length (X) of bytes following	Y
Y+2 to X+Y+1	Event list	X

Event list:

- Contents:
  - A list of events, of variable length. Each byte in the list defines an event. Each event type shall not appear more than once within the list.
- Coding:
  - Each byte in the event list shall be coded with one of the values below:
    - '00' = MT call;
    - '01' = Call connected;
    - '02' = Call disconnected;
    - '03' = Location status;
    - '04' = User activity;
    - '05' = Idle screen available;
    - '06' = Card reader status;
    - '07' = Language selection;
    - '08' = Browser termination;
    - '09' = Data available;
    - '0A' = Channel status;
    - '0B' = Access Technology Change (single access technology);
    - '0C' = Display parameters changed;
    - '0D' = Local connection;
    - '0E' = Network Search Mode Change;
    - '0F' = Browsing status;

- '10' = Frames Information Change;
- '11' = reserved for 3GPP (I-WLAN Access Status);
- '12' = reserved for 3GPP (Network Rejection);
- '13' = HCI connectivity event;
- '14' = Access Technology Change (multiple access technologies);
- '15' = reserved for 3GPP (CSG cell selection);
- '16' = Contactless state request;
- '17' = reserved for 3GPP (IMS Registration);
- '18' = reserved for 3GPP (IMS Incoming data);
- '19' = Profile Container;
- '1A' = Void;
- '1B' = Secured Profile Container.

## 8.26 Cause

Byte(s)	Description	Length
1	Cause tag	1
2	Length (X) of bytes following. $X=0$ , or $2 \leq X \leq 30$	1
3 to X+2	Cause	X

The cause value is defined in the appropriate access technology specification. Radio Link Timeout is indicated by the Cause data object having a value part of zero length (only the Tag and Length components are sent).

## 8.27 Location status

Byte(s)	Description	Length
1	Location status tag	1
2	Length (X) of bytes following	1
3	Location status	1

Location status:

- Contents:
  - This data object indicates the current service state of the terminal:
    - "normal service" shall indicate that the terminal is in a state where all requests for services are treated normally;
    - "limited service" shall indicate that the terminal is in a state where only emergency call services are offered;
    - "no service" shall indicate that the terminal is in a state where no services are offered.
- Coding:
  - Each byte in the event list shall be coded with one of the values below:
    - '00' = Normal service;
    - '01' = Limited service;

- '02' = No service.

## 8.28 Transaction identifier

Coding according to specific technology.

## 8.29 Void

## 8.30 Call control requested action

Byte(s)	Description	Length
1	Call control requested action tag	1
2 to (Y-1)+2	Length (X)	Y
(Y-1)+3 to (Y-1)+X+2	Call control requested action	X

- Contents:
  - The action given in response to the ENVELOPE (CALL CONTROL).
- Coding:
  - As described in clause 7.3.1.6, starting with the first optional element given in the response data to the ENVELOPE (CALL CONTROL). The TLV elements shall be in the same order as given by the UICC.

## 8.31 Icon identifier

Byte(s)	Description	Length
1	Icon identifier tag	1
2	Length = '02'	1
3	Icon qualifier	1
4	Icon identifier	1

Icon qualifier:

- Contents:
  - The icon qualifier indicates to the terminal how the icon is to be used.
- Coding:
  - bit 1: 0 = icon is self-explanatory, i.e. if displayed, it replaces the alpha identifier or text string;  
1 = icon is not self-explanatory, i.e. if displayed, it shall be displayed together with the alpha identifier or text string.
  - bits 2 to 8 = 0 RFU.

Icon identifier:

- Contents:
  - The icon identifier addresses a record in EF<sub>IMG</sub> as defined in TS 131 102 [6].
- Coding:
  - Binary.

## 8.32 Item icon identifier list

Byte(s)	Description	Length
1	Items Icon identifier tag	1
2	Length (X) of bytes following	1
3	Icon list qualifier	1
4 to 4+X-2	Icon identifier list	X-1

Icon list qualifier:

- Contents:
  - The icon list qualifier indicates to the terminal how the icons are to be used.
- Coding:
  - bit 1: 0 = icon is self-explanatory, i.e. if displayed, it replaces the item text;
    - 1 = icon is not self-explanatory, i.e. if displayed, it shall be displayed together with the item text.
  - bits 2 to 8 = 0 RFU.

All icons in the list shall be treated in the same manner by the terminal, i.e. either none of the icons in this list are displayed, or for each item its related icon is displayed.

Icon identifier list:

- Contents:
  - Each item of a list of items has an icon identifier coded on one byte. The length of the Items icon identifier list shall be the number of items of the list of items (X-1 shall be the number of items in the list). The order of each item icon identifier shall reflect the order of the items in the list of items.
  - Each icon identifier addresses a record in EF<sub>IMG</sub> as defined in TS 131 102 [6].
- Coding: Binary.

EXAMPLE: For the following list of items:

- item #1;
- item #2;
- item #3;
- ....;
- item #n.

The Items icon identifier list shall be as follows:

Tag	Length	Icon list qualifier	icon identifier#1	icon identifier#2	icon identifier#3	...	icon identifier #n

## 8.33 Card reader status

Byte(s)	Description	Length
1	Card reader status tag	1
2	Length	1
3	Card reader status	1

- Contents:
  - This contains the identity of the additional card reader, and flags to indicate the status of the reader with respect to:
    - whether the card reader is removable or permanently connected;
    - whether the card reader is present (this can only be false if the card reader is removable);
    - whether the card reader present accepts ID-1 size cards (this can only be true if the card reader is present);
    - whether there is a card present in the card reader (this can only be true if the card reader is present);
    - whether power is being applied to the card (this can only be true if a card is present).
- Coding:
  - The value of this byte indicates the identity and status of a card reader:
    - bits 1 to 3 = Identity of the additional card reader x (decimal 0 to 7) as assigned by the terminal.
    - bit 4        0 = Card reader is not removable;  
                  1 = Card reader is removable.
    - bit 5        0 = Card reader is not present;  
                  1 = Card reader is present.
    - bit 6        0 = Card reader present is not ID-1 size;  
                  1 = Card reader present is ID-1 size.
    - bit 7        0 = No card present;  
                  1 = Card is present in reader.
    - bit 8        0 = No card powered;  
                  1 = Card in reader is powered.

## 8.34 Card ATR

Byte(s)	Description	Length
1	Card ATR tag	1
2	Length (X) of bytes following	1
3 to (X+2)	ATR	X

ATR:

- Contents:
  - This is the Answer To Reset returned by the card.
- Coding:
  - The coding of the Answer To Reset is defined in ISO/IEC 7816-3 [13].



## 8.35 C-APDU

Byte(s)	Description	Length
1	C-APDU tag	1
2 to (Y+1)	Length (X) of bytes following (Y = 1 or 2)	Y
Y+2	Command class CLA	1
Y+3	Command instruction code INS	1
Y+4	P1 parameter	1
Y+5	P2 parameter	1
Y+6	Lc (optional)	0 or 1
(Y+7) to (Y+X)	Data (optional)	Lc
Y+X+1	Le (optional)	0 or 1

This object contains the command APDU for Card x in the format defined in ISO/IEC 7816-4 [14]. Command class CLA, instruction code INS, P1 and P2 parameters, Lc, Data and Le are coded as defined in ISO/IEC 7816-4 [14]. Extended lengths are not supported.

NOTE: The maximum size of the value part of this COMPREHENSION-TLV (value of X) is limited to 241 bytes, so the maximum length for the Data (value of Lc) in a Case 3 type of APDU is 236 bytes.

## 8.36 R-APDU

Byte(s)	Description	Length
1	R-APDU tag	1
2 to Y+1	Length (X) of bytes following (Y = 1 or 2)	Y
Y+2 to Y+X-1	R-APDU data (optional)	X-2
Y+X	Status word SW1	1
Y+X+1	Status word SW2	1

This object contains the response APDU from Card x in the format defined in ISO/IEC 7816-4 [14]. The R-APDU data and status words SW1 and SW2 are coded as defined in ISO/IEC 7816-4 [14]. It is possible for no R-APDU data to be present; this is indicated by the length of the data object.

NOTE: The maximum size of the value part of this COMPREHENSION-TLV (value of X) is limited to 239 bytes, so the maximum length of the R-APDU data is 237 bytes.

## 8.37 Timer identifier

Byte(s)	Description	Length
1	Timer identifier tag	1
2	Length='01'	1
3	Timer identifier	1

Timer identifier:

- Contents: identifier of a timer.
- Coding:
  - '01' Timer 1;
  - '02' Timer 2;
  - '03' Timer 3;
  - '04' Timer 4;
  - '05' Timer 5;

- '06' Timer 6;
- '07' Timer 7;
- '08' Timer 8;
- All other values are reserved.

## 8.38 Timer value

Byte(s)	Description	Length
1	Timer value tag	1
2	Length='03'	1
3 to 5	Timer value	3

Timer value:

- Contents:
  - value of a timer, expressed using the format hour, minute, second.
- Coding:
  - byte 3: hour; this byte is coded exactly in the same way as the hour field of the TP-Service-Centre-Time-Stamp in TS 123 040 [27];
  - byte 4: minute; this byte is coded exactly in the same way as the minute field of the TP-Service-Centre-Time-Stamp in TS 123 040 [27];
  - byte 5: second; this byte is coded exactly in the same way as the second field of the TP-Service-Centre-Time-Stamp in TS 123 040 [27].

## 8.39 Date-time and time zone

Byte(s)	Description	Length
1	Date-Time and Time zone tag	1
2	Length = '07'	1
3 to 9	Date-Time and Time zone	7

Date-Time and Time Zone:

- Contents:
  - Date, Time and Time Zone.
- Coding:
  - byte 3: Year; this byte is coded exactly in the same way as the Year field of the TP-Service-Centre-Time-Stamp in TS 123 040 [27];
  - byte 4: Month; this byte is coded exactly in the same way as the Month field of the TP-Service-Centre-Time-Stamp in TS 123 040 [27];
  - byte 5: Day; this byte is coded exactly in the same way as the Day field of the TP-Service-Centre-Time-Stamp in TS 123 040 [27];
  - byte 6: Hour; this byte is coded exactly in the same way as the Hour field of the TP-Service-Centre-Time-Stamp in TS 123 040 [27];
  - byte 7: Minute; this byte is coded exactly in the same way as the Minute field of the TP-Service-Centre-Time-Stamp in TS 123 040 [27];

- byte 8: Second; this byte is coded exactly in the same way as the Second field of the TP-Service-Centre-Time-Stamp in TS 123 040 [27];
- byte 9: Time Zone; this byte is coded exactly in the same way as the Time Zone field of the TP-Service-Centre-Time-Stamp in TS 123 040 [27]. 'FF' indicates an unknown value.

## 8.40 AT command

Byte(s)	Description	Length
1	AT Command tag	1
2 to (Y-1)+2	Length (X)	Y
(Y-1)+3 to (Y-1)+3+X-1	AT Command string	X

- Contents:
  - The AT Command string is structured exactly as the AT Command line as defined in TS 127 007 [5], which may contain single or concatenated AT commands. Each NAA may have its specific AT command specification.

## 8.41 AT response

Byte(s)	Description	Length
1	AT Response tag	1
2 to (Y-1)+2	Length (X)	Y
(Y-1)+3 to (Y-1)+3+X-1	AT Response string	X

- Contents:
  - The AT Response string is structured exactly as the response to a command line as defined in TS 127 007 [5], which may contain single or concatenated responses appropriate to the issued AT command. Each NAA may have its specific AT command specification.
  - If the AT Response string is longer than the maximum length capable of being transmitted to the UICC then the AT Response string shall be truncated to this length by the terminal.

## 8.42 BC repeat indicator

Byte(s)	Description	Length
1	BC repeat indicator tag	1
2	Length	1
3	BC repeat indicator values	1

BC repeat indicator coding is defined by the different access technologies.

## 8.43 Immediate response

This TLV object is used in the sustained DISPLAY TEXT and in the sustained DISPLAY MULTIMEDIA MESSAGE commands.

Byte(s)	Description	Length
1	Immediate response tag	1
2	Length='00'	1

## 8.44 DTMF string

Byte(s)	Description	Length
1	DTMF String tag	1
2 to (Y-1)+2	Length (X)	Y
(Y-1)+3 to (Y-1)+3+X-1	DTMF string	X

- Contents:
  - The DTMF string which can be single or multiple characters is coded in BCD, in the same way as the Dialling number string defined for EF<sub>ADN</sub> in TS 131 102 [6]. It may include extended BCD coding. There is no need for a DTMF control digit separator at the beginning of the string, but if present it shall be interpreted as PAUSE.

## 8.45 Language

Byte(s)	Description	Length
1	Language tag	1
2	Length = '02'	1
3 to 4	Language	2

- Coding:
  - Each language code is a pair of alpha-numeric characters, defined in ISO 639 [12]. Each alpha-numeric character shall be coded on one byte using the SMS default 7-bit coded alphabet as defined in TS 123 038 [3] with bit 8 set to 0.

## 8.46 Void

## 8.47 Browser identity

Byte(s)	Description	Length
1	Browser identity tag	1
2	Length	1
3	Browser Identity	1

- Coding:
  - 00 = Default Browser shall be used;
  - 01 = WML Browser shall be used;
  - 02 = HTML Browser shall be used;
  - 03 = XHTML Browser shall be used;
  - 04 = CHTML Browser shall be used;
  - Other values are RFU.

## 8.48 URL

Byte(s)	Description	Length
1	URL tag	1
2 to (Y+1)	Length (X)	Y
(Y+2) to (Y+1+X)	URL	X

A null URL shall be coded with Length = '00', and no Value part. In that case, the terminal shall use the default URL.

- Coding:
  - The data used for the URL shall be coded as defined in RFC 1738 [11] on using the "SMS 7bit default alphabet" with bit 8 set to 0.

## 8.49 Bearer

Byte(s)	Description	Length
1	Bearer tag	1
2 to (Y+1)	Length (X)	Y
(Y+2) to (Y+X+1)	List of bearers in order of priority requested	X

The terminal shall use this list to choose which bearers are allowed in order of priority.

- Coding:
  - '00' = short message;
  - '01' = circuit switched data;
  - '02' = reserved for GSM/3G;
  - '03' = packet switched;
  - '04' to 'FF' = RFU.

## 8.50 Provisioning file reference

Byte(s)	Description	Length
1	Provisioning file reference tag	1
2 to (Y+1)	Length (X)	Y
(Y+2) to (Y+X+1)	Path to the provisioning file	X

NOTE: The path is the concatenation of file identifiers starting from the Master File, e.g. 3F007F106FX Y, etc.

The file shall contain a single unambiguous set of parameters required to make the connection. The content of the file shall be consistent with the format defined for provisioning information for the requested type of browser.

## 8.51 Browser termination cause

Byte(s)	Description	Length
1	Browser Termination Cause tag	1
2	Length (1)	1
3	Browser Termination Cause	1

- Coding:
  - 00 = User Termination;
  - 01 = Error Termination.

## 8.52 Bearer description

Byte(s)	Description	Length
1	Bearer description tag	1
2	Length (X+1)	1
3	Bearer type	1
4 to (3+X)	Bearer parameters	X

- Coding of the Bearer Type:
  - '01' = reserved for GSM/3GPP;
  - '02' = reserved for GSM/3GPP;
  - '03' = default bearer for requested transport layer;
  - '04' = local link technology independent;
  - '05' = Bluetooth;
  - '06' = IrDA;
  - '07' = RS232;
  - '08' = TIA/EIA/IS-820-A [17] packet data service;
  - '09' = reserved for GSM/3GPP;
  - '0A' = reserved for 3GPP (I-WLAN);
  - '0B' = reserved for 3GPP (E-UTRAN/Mapped UTRAN packet service);
  - '10' = USB;
  - All other values are reserved.
- Coding of the Bearer Parameters:
  - default bearer:
    - the terminal shall provide its default available bearer parameter configuration. X (length of parameters) = 0;
  - local links (Bluetooth, IrDA, RS232, USB):
    - in this case, X = variable. Contains "Service Identifier" and "Service Record" fields as defined in clause 8.63 and according to the Bearer Type coding;
  - other bearers:
    - the coding is defined in the appropriate access technology specific specification.

## 8.53 Channel data

Byte(s)	Description	Length
1	Channel data tag	1
2 to Y+1	Length (X)	Y
(Y+2) to (Y+X+1)	Channel data string	X

- Contents:
  - The Channel data object contains application data read from or written to a specific channel buffer in the terminal.
- Coding:
  - The Channel data string shall be considered by the terminal as binary coded on 8 bits.

## 8.54 Channel data length

Byte(s)	Description	Length
1	Channel data length tag	1
2	Length (1)	1
3	Channel data length	1

- Contents:
  - Either the number of bytes that are available in a channel buffer (Tx or Rx buffers negotiated during OPEN CHANNEL) using TERMINAL RESPONSE. Since the Tx or Rx buffer size can be larger than 255 bytes, 'FF' means "more than 255 bytes are available".
  - Or the number of bytes that are requested in a RECEIVE DATA command.

## 8.55 Buffer size

Byte(s)	Description	Length
1	Buffer size tag	1
2	Length (2)	1
3 to 4	Buffer size	2

- Contents:
  - The Buffer size codes the number of bytes requested by the UICC in an OPEN CHANNEL command or what the terminal can offer the UICC (placed in TERMINAL RESPONSE).

## 8.56 Channel status

Byte(s)	Description	Length
1	Channel status tag	1
2	Length (2)	1
3 to 4	Channel status	2

- Contents:
  - The Channel status is a string of binary coded characters.

- Coding:

- byte 3:

- bit 1 to 3: Channel identifier: 1 to 7;  
 Channel identifier 0 means "No channel available".

For CS, packet data service, local and Default (network) bearer:

- bit 4 to 7: RFU.
- bit 8: 0 = Link not established or Packet data service not activated;  
 1 = Link established or Packet data service activated.

For UICC Server Mode:

- bit 4 to 6: RFU.
- bit 7, 8: 00 = TCP in CLOSED state;  
 01 = TCP in LISTEN state;  
 10 = TCP in ESTABLISHED state;  
 11 = reserved.

For Terminal Server Mode and TCP or direct communication channel:

- bit 4 to 6: RFU.
- bit 7, 8: 00 = TCP in CLOSED state/direct communication channel closed;  
 01 = reserved;  
 10 = TCP in ESTABLISHED state/direct communication channel established;  
 11 = reserved.

For Terminal Server Mode and UDP:

- bit 4 to 8: RFU.

- byte 4:

- '00' = No further info can be given;
- '01' = Not used;
- '02' = Not used;
- '03' = Not used;
- '04' = Not used;
- '05' = Link dropped (network failure or user cancellation);
- all other values are reserved.

## 8.57 Card reader identifier

Byte(s)	Description	Length
1	Card reader identifier tag	1
2	Length (X)	1
3 to (X+2)	Identifier of card reader	X



Identifier of card reader:

- Contents:
  - This contains manufacturer-specific information to identify the type of card reader being used.
- Coding:
  - The identifier of card reader is coded in hexadecimal.

## 8.58 Other Address

Byte(s)	Description	Length
1	Other address tag	1
2	Length (X)	1
3	Type of address	1
4 to (X+2)	Address	X-1

A null Local address shall be coded with Length = '00', and no Value part. In that case, the terminal shall request a dynamic address; the type of address and the address shall be provided by the terminal (placed in TERMINAL RESPONSE).

- Coding of Type of address:
  - '21' = IPv4 address;
  - '57' = IPv6 address;
  - 'others' = reserved.
- Coding of address:
  - If type of address indicates IPv4, the Address information in octet 4 to octet 7 contains the IPv4 address. Bit 8 of octet 4 represents the most significant bit of the IP address and bit 1 of octet 7 the least significant bit.
  - If type of address indicates IPv6, the Address information in octet 4 to octet 19 contains the IPv6 address. Bit 8 of octet 4 represents the most significant bit of the IP address and bit 1 of octet 19 the least significant bit.

## 8.59 UICC/terminal interface transport level

This clause applies if class "e" is supported.

Byte(s)	Description	Length
1	UICC/terminal interface transport level tag	1
2	Length = "03"	1
3	Transport protocol type	1
4 to 5	Port number	2

- Coding of the Transport protocol type:
  - '01': UDP, UICC in client mode, remote connection;
  - '02': TCP, UICC in client mode, remote connection;
  - '03': TCP, UICC in server mode;
  - '04': UDP, UICC in client mode, local connection;
  - '05': TCP, UICC in client mode, local connection;

- '06': direct communication channel;
- all other values are reserved.
- UDP is defined in RFC 768 [9]; TCP is defined in RFC 793 [10].
- Coding of the Port number:
  - integer.

## 8.60 AID

Byte(s)	Description	Length
1	AID tag	1
2	Length (X)	1
3 to (X+2)	AID	X

- Contents:
  - Application identifier as defined in TS 101 220 [31].

## 8.61 Access technology

Byte(s)	Description	Length
1	Access Technology tag	1
2	Length (X) of bytes following	1
3	Technology	X

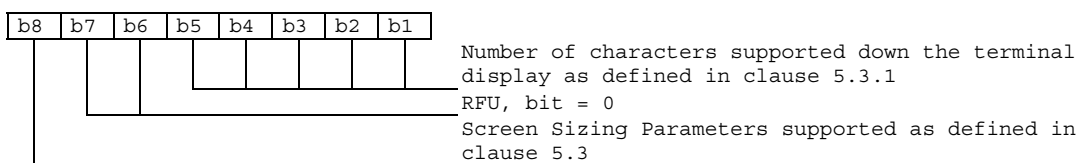
- Contents:
  - The terminal shall use this information as a mechanism to indicate to the UICC the current access technology that it is using.
- Coding:
  - '00' = GSM;
  - '01' = TIA/EIA-553 [49];
  - '02' = TIA/EIA-136-C [25];
  - '03' = UTRAN;
  - '04' = TETRA;
  - '05' = TIA/EIA-95 [50];
  - '06' = cdma2000 1x (TIA/EIA/IS-2000 [51]);
  - '07' = cdma2000 HRPD (TIA/EIA/IS-856 [52]);
  - '08' = E-UTRAN;
  - All other values are reserved for future use.

## 8.62 Display parameters

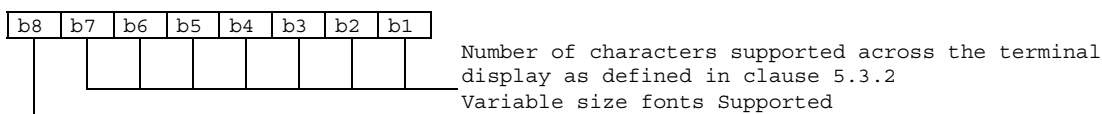
Byte(s)	Description	Length
1	Display parameters tag	1
2	Length = '03'	1
3 to 5	Parameters list	3

- Contents:
  - A list of different information regarding the terminal's screen.
- Coding:
  - One bit is used to code parameters supported or not:
    - bit = 1: parameters supported by terminal;
    - bit = 0: parameters not supported by terminal.

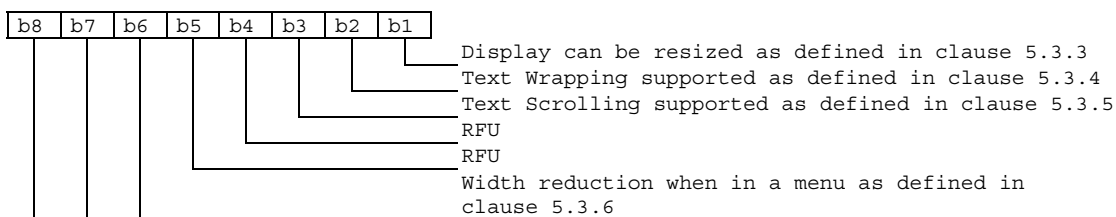
First byte (Screen height):



Second byte (Screen width):



Third byte (Screen effects):



## 8.63 Service record

This service record can have different formats that are dependent on the technology they are associated with.

This object can be used in both directions (terminal to UICC or UICC to terminal), when a CAT application needs to declare a service that it supports (DECLARE SERVICE command) and when CAT application searches for a service (GET SERVICE INFORMATION).

Byte(s)	Description	Length
1	Service Record tag	1
2 to Y+1	Length (X+2)	Y
Y+2	Local Bearer technology identifier	1
Y+3	Service Identifier	1
Y+4 to Y+X+3	Service Record	X

Local Bearer Technology identifier:

- Coding:
  - '00' = Technology independent: '00';
  - '01' = Bluetooth;
  - '02' = IrDA;
  - '03' = RS232;
  - '04' = USB;
  - '05' to 'FF' = RFU.

Service identifier:

- Coding:
  - When declaring a service, the UICC associates a Service Identifier to the Service Record. When the Service Record TLV is returned in response to GET SERVICE INFORMATION, Service Identifier shall be set to 'FF'.
    - '00' to '07' Service x (0 to 7). Value assigned by the UICC;
    - 'FF' = Service Record related to the service provided by a remote device.

Other value reserved for future use.

Service Record:

When the Service Record field is not meaningful, it shall be assigned the value = '00'.

- Technology Independent:
  - RFU.
- Bluetooth:
  - In Bluetooth, a Service record gives all needed information that shall be used by a device to connect and use this service. The full description of the coding of these records is given in the Bluetooth Specification in the SDP clause [16]. When Service Record is returned in response to GET SERVICE INFORMATION, it corresponds to the AttributeList parameter contained in the SDP\_ServiceAttributeResponse PDU [16].
- IrDA:
  - In IrDA a Service record gives all needed information that shall be used by a device to connect and use this service.

The full description of the coding of these records is given in the IrLMP specification [33] in the Information Access Service clause. When Service Record is returned in response to GET SERVICE INFORMATION, it corresponds to the results of series of LM\_GetValueByClass operation. The operation LM\_GetValueByClass enables the client to access all the values of a named attribute in objects of a given class name. This service is required in all IAS servers. The terminal shall repeatedly call the LM\_GetValueByClass operation to get the values of the attributes defined in the Attribute Information TLV.

When Service Record is used with DECLARE SERVICE or OPEN CHANNEL it corresponds to the class name, followed by couples of attribute name and attribute value.
- RS232:
  - RFU.

- USB:
  - RFU.

Depending on the proactive command, the parameters of this TLV could be either meaningful or optional. The following table indicates in which case the parameters are required.

Proactive command	Service Identifier required	Service Record field required
DECLARE SERVICE (add)	Yes	Yes
DECLARE SERVICE (delete)	Yes	No (value '00' assigned)
terminal response of a GET SERVICE INFORMATION	No (value 'FF' assigned)	Yes
OPEN CHANNEL (client)	No (value 'FF' assigned)	Yes
OPEN CHANNEL (server)	Yes	No (value '00' assigned)
Local Connection event	Yes	No (value '00' assigned)

## 8.64 Device filter

Byte(s)	Description	Length
1	Device Filter tag	1
2 to Y+1	Length (1+X1+X2+...+Xn)	Y
Y+2	Local Bearer technology identifier	1
Y+3 to Y+2+X	Device Filter	X

Local Bearer Technology identifier: see clause 8.63.

Device filter:

If the Local Bearer Technology Identifier is different from '00', the device filter coding is technology dependent.

- Technology Independent:
  - RFU.
- Bluetooth:
  - The Device Filter parameter is used to filter the responses to a service search. For Bluetooth, it is a list of Class\_Of\_Device and Class\_Of\_Device\_Mask.
  - Device Filter =
    - Class\_Of\_Device\_1 [3 bytes], Class\_Of\_Device\_Mask\_1 [3 bytes];
    - Class\_Of\_Device\_2 [3 bytes], Class\_Of\_Device\_Mask\_2 [3 bytes];
    - ...;
    - Class\_Of\_Device\_n [3 bytes], Class\_Of\_Device\_Mask\_n [3 bytes].
- IrDA:
  - The device Filter Parameter is used to limit service search to a set of devices. For IrDA, it is the Service Hints bytes (the first n bytes of the DeviceInfo field). Service hints should not be taken to mean a particular service is provided by the device. They are merely to provide assistance in choosing a set of device to contact during the discovery process. The full description of the Service Hints bytes is given in the IrLMP specification [33] in the "Frame Formats" clause.
  - Device Filter = Service Hints

The eighth bit of every hint byte (bit 7, 15, 23, etc.) is an extension bit and indicates whether or not an additional hint byte is included.

- RS232:
  - RFU.
- USB:
  - RFU.

## 8.65 Service search

Byte(s)	Description	Length
1	Service Search tag	1
2 to Y+1	Length (X+1)	Y
Y+2	Local Bearer technology identifier	1
Y+3 to Y+X+1	Service Search	X

Local Bearer Technology identifier: see clause 8.63.

Service search:

If the Local Bearer Technology Identifier is different from '00', the Service search coding is technology dependent.

- Technology Independent:
  - RFU.
- Bluetooth:
  - The Service Search field is the *ServiceSearchPattern* parameter of the *SDP\_ServiceSearchRequest* command as defined in the Bluetooth specification [16].
- IrDA:
  - The Service Search field is the class name parameter of the IAS LM\_GetValueByClass command as defined in [33].
  - The terminal shall perform an IAS LM\_GetValueByclass.request call using the class name given in Service Search, and check the return code value of the associated IAS LM\_GetValueByclass.confirm. Only devices, which respond with a return code different from 1 (no such class), implement the service.
  - Alternatively, since the LSAP selector is mandatory in all IAS classes, the terminal can perform two LM\_GetValueByClass calls using "IrDA:TinyTP:LsapSel" and "IrDA:IrLMP:LsapSel" as parameters values. If the terminal receives one of those parameters value, it concludes that the service exists.
- RS232:
  - RFU.
- USB:
  - RFU.

## 8.66 Attribute information

Byte(s)	Description	Length
1	Item tag	1
2 to Y+1	Length (X+1)	Y
Y+2	Local Bearer technology identifier	1
Y+3 to Y+X+2	Attribute Information	X

Local Bearer Technology identifier: see clause 8.63.

## Attribute Information:

If the Local Bearer Technology Identifier is different from '00', the Attribute Information coding is technology dependent.

- Technology Independent:
  - RFU.
- Bluetooth:
  - The Attribute Information field consists of a *BD\_ADDR*, followed by the *ServiceRecordHandle* and the *AttributeIDList* parameters of the *SDP\_ServiceAttributeRequest* command as defined in the Bluetooth specification [16].
  - The *BD\_ADDR* is the Bluetooth device address of the device the terminal shall connect to. The terminal shall use the *ServiceRecordHandle* and the *AttributeIDList* parameters to perform the *SDP\_ServiceAttributeRequest*. The *ServiceRecordHandle* has been previously retrieved with the *SERVICE SEARCH* command.
- IrDA:
  - The Attribute Information field consists of the device address the terminal shall connect to, followed by the class name and one or several attributes names. The full description of the coding of these records is given in the IrLMP specification [33].
  - The terminal shall use the device address, the class name, and the attributes names to perform series of *IAS LM\_GetValueByClass* call in order to access attributes values.
- RS232:
  - RFU.
- USB:
  - RFU.

## 8.67 Service availability

The Service Availability parameter contains a list of available services that the *SERVICE SEARCH* command returns. This object is formatted according to the local bearer technology identifier byte set in the *SERVICE SEARCH* command arguments.

Byte(s)	Description	Length
1	Service Availability tag	1
2 to Y+1	Length='X1'+ 'X2'+ 'X3'+... 'Xn' (n maxi = 7)	Y
Y+2 to Y+X1+1	Service_1	X1
Y+X1+2 to Y+X1+X2+1	Service_2	X2
...	...	...
Y+X1+...+X(n-1)+2 to Y+X1+...+Xn+1	Service_n	Xn

- Technology Independent:
  - RFU.

- Bluetooth:
  - For Bluetooth,  $Service\_i = BD\_ADDR\_i[6 \text{ bytes}] + ServiceRecordHandle\_i[4 \text{ bytes}] + CoD\_i[3 \text{ bytes}] + Device\_Name\_i[20 \text{ bytes}]$ , those parameters being defined in the Bluetooth specification [16].

Byte(s)	Description	Length
1	Service Availability tag	1
2 to Y+1	Length='X1'+ 'X2'+ 'X3'+... 'Xn' (n maxi = 7)	Y
Y+2 to Y+X1+1	BD_ADDR + ServiceRecordHandle + CoD + Device_Name	X1
Y+X1+2 to Y+X1+X2+1	BD_ADDR + ServiceRecordHandle + CoD + Device_Name	X2
...	...	...
Y+X1+...+X(n-1)+2 to Y+X1+...+Xn+1	BD_ADDR + ServiceRecordHandle + CoD + Device_Name	Xn

- IrDA:
  - For IrDA,  $Service\_i = adress\_i[4 \text{ bytes}] + Character\_Set\_I[1 \text{ byte}] + ServiceHint\_i[2 \text{ bytes}] + Device\_NickName\_i[20 \text{ bytes}]$ , those parameters being defined in [33].

Byte(s)	Description	Length
1	Service Availability tag	1
2 to Y+1	Length='X1'+ 'X2'+ 'X3'+... 'Xn' (n max = 7)	Y
Y+2 to Y+X1+1	address + character set+ ServiceHints + Device_NickName	X1
Y+X1+2 to Y+X1+X2+1	address + character set+ ServiceHints + Device_NickName	X2
...	...	...
Y+X1+...+X(n-1)+2 to Y+X1+...+Xn+1	address + character set+ ServiceHints + Device_NickName	Xn

NOTE: If the device nick name field is empty the client can find out the device name by querying the mandatory Object 0 (instance of class device). The client can call the GetValueByClass operation setting the class name parameter to 'Device' and the attribute name parameter to 'DeviceName').

- RS232:
  - RFU.
- USB:
  - RFU.

## 8.68 Remote entity address

Byte(s)	Description	Length
1	Remote Entity Address tag	1
2 to Y+1	Length (X+1)	Y
Y+2	Coding Type	1
Y+3 to Y+X+2	Remote Entity address	X

- Coding Type:
  - '00': IEEE-802 [46] 48-bit address;
  - '01': 32 bits IrDA device address;
  - '02' to 'FF' are reserved values.
- Remote Entity Address:
  - according to Coding Type.



## 8.69 ESN

Byte(s)	Description	Length
1	ESN tag	1
2	Length = '04'	1
3 to 6	ESN of the terminal	4

The ESN is coded as in TIA/EIA-41-D [18].

## 8.70 Network access name

Byte(s)	Description	Length
1	Network Access Name tag	1
2	Length (X)	1
3 to 3+X-1	Network Access Name	X

- Content:
  - The Network Access Name is used to identify the Gateway entity, which provides interworking with an external packet data network.
- Coding:
  - Defined by the different access technologies.

## 8.71 CDMA-SMS TPDU

Contents and coding: 3GPP2 C.S0015-0 [29].

## 8.72 Text attribute

Byte(s)	Description	Length
1	Text Attribute Tag	1
2	Length (X)	1
3 to X+2	Text Formatting	X

- Text Formatting:
  - Coding:
    - The Text Formatting is a sequence of one or several Text Formatting items, each coded on 4 bytes. The Text Formatting scheme used is the same as the Text Formatting defined in TS 123 040 [27].

## 8.73 Item text attribute list

Byte(s)	Description	Length
1	Item Text Attribute List tag	1
2	Length (X)	1
3 to (2+X)	Text Attribute list	X

The Item text attribute list comprehension TLV contains a list of Text Attributes. All Text Attributes in the list shall be treated in the same manner by the terminal, i.e. either none of the Text Attributes in this list are displayed, or for each item its related text format is displayed.

- Text Attribute list:
  - Contents:
    - Each item of the list is a Text Attribute coded on 4 bytes. The text formatting scheme used for the Text Attribute is the same as the Text Formatting defined in TS 123 040 [27]. The length of the Text Attribute list shall be the number of items of the list multiplied by 4. The order of each Text Attribute, shall reflect the order of the item in the list.

EXAMPLE: For the following list of items:

- item #1;
- item #2;
- item #3;
- ....;
- item #n.

The Item text attribute list shall be coded as follows.

Tag	Length	Text Attribute #1	Text Attribute #2	Text Attribute #3	...	Text Attribute #n
-----	--------	-------------------	-------------------	-------------------	-----	-------------------

## 8.74 IMEISV

Byte(s)	Description	Length
1	IMEISV tag	1
2	Length (X)	1
3 to (X+2)	IMEISV of the terminal	X

The IMEISV is coded in the same manner as the value part of the Mobile Identity information element as specified in TS 124 008 [20]. The IMEISV itself is specified in TS 123 003 [32].

## 8.75 Network search mode

Byte(s)	Description	Length
1	Network search mode tag	1
2	Length	1
3	Network search mode	1

Network type selection:

- Coding:
  - '00' = Manual;
  - '01' = Automatic;
  - '02' to 'FF' = RFU.

## 8.76 Battery State

This clause applies only if class "g" is supported.

Byte(s)	Description	Length
1	Battery State tag	1
2	Length = '01'	1
3	Battery State	1

- Contents:
  - The Battery State data object reflects the current charge state of the (rechargeable) battery in the Terminal.
- Coding:
  - '00' = battery very low;
  - '01' = battery low;
  - '02' = battery average;
  - '03' = battery good;
  - '04' = battery full;
  - '05' to 'FD' = RFU;
  - 'FE' = Status not applicable - Not powered by a Battery;
  - 'FF' = Status Unknown - e.g. battery is charging.

## 8.77 Browsing status

Byte(s)	Description	Length
1	Browsing status tag	1
2	Length (X)	Y
3	Browsing status	X

- Coding:
  - The browsing status contains the error code sent by the network and received by the browser.

## 8.78 Frame Layout

Byte(s)	Description	Length
1	Frame Layout tag	1
2	Length = '1+X'	1
3	Layout of the frames	1
4 to 4+X	Relative-sized Frame	X

Layout of the frames:

- Contents:
  - The layout of frames: horizontal or vertical.

- Coding:
  - '01' = horizontal frames;
  - '02' = vertical frames.

Relative size of the new frames:

- Contents:
  - One byte for each frame to be created. The content of each byte defines the relative frame size. If all bytes have the same value, the screen/frame shall be split into equal sized (sub-)frames. If 2 bytes are different, the values define the relative size of the frames. For instance "2,1" would give 2/3 of the space to the first frame, and 1/3 to the second.
- Coding:
  - '01'..'FF' = Relative size of the frame.

## 8.79 Frames Information

Byte(s)	Description	Length
1	Frame Information tag	1
2	Length = 'X+1'	1
3	Default Frame ID	1
4 to 3+X	Frame Information List	X

Default Frame ID:

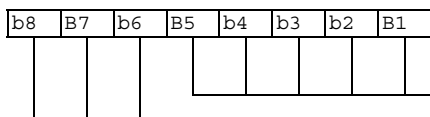
- Contents:
  - This ID is used by the proactive commands without frame identifier to select the frame to use.
- Coding:
  - Any value between '01' to '0F';
  - '00': if no frames are defined;
  - '10' to FF: Reserved values.

Frame Information List:

- Contents:
  - A list of different information regarding the terminal's current frames. The length shall be double the number of frames. It is empty if no frames are currently defined. The  $2*N^{th}-1$  and the  $2*N^{th}$  byte correspond to the  $N^{th}$  Frame.

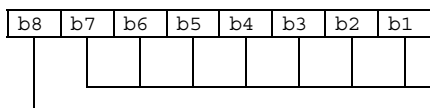
- Coding:

- $2*N^{th}-1$  byte:



Number of characters supported down the  $N^{th}$  Frame display as defined in 5.3.1  
RFU, bit = 0

- $2*N^{th}$  byte:



Number of characters supported across the terminal display as defined in clause 5.3.2  
RFU, bit = 0

## 8.80 Frame identifier

Byte(s)	Description	Length
1	Frame identifier tag	1
2	Length = '01'	1
3	Identifier of frame	1

The identifier is a single byte between '00' and '0F', exactly the same as for the Item data object.

The value '00' is reserved to mean entire terminal screen, and is only valid as frame identifier in the Set Up Frames command.

- '10' to 'FF': RFU.

## 8.81 MEID

Byte(s)	Description	Length
1	MEID tag	1
2	Length = '08'	1
3 to 10	See 3GPP2 C.S0035-A [47]	8

## 8.82 Multimedia Message Reference

This clause applies if class "j" is supported.

Byte(s)	Description	Length
1	Multimedia Message Reference tag	1
2	Length (X)	1
3	Multimedia Message Reference	X

Multimedia Message Reference:

- Contents:
  - this contains Multimedia Message Reference used to retrieve the MM from the network;
- Coding:
  - the Multimedia Message Reference is the "MM1\_retrieve.REQ", see TS 123 140 [37] for further details.

## 8.83 Multimedia Message Identifier

This clause applies if class "j" is supported.

Byte(s)	Description	Length
1	Multimedia Message Identifier tag	1
2	Length (X)	1
3	Multimedia Message Identifier	X

Identifier of Multimedia Message:

- Contents:
  - this contains Multimedia Message Identifier to be used to retrieve a Multimedia Message. This identifier is mandatory in case the MMS Reception or Submission file can store several MMs;

- Coding:
  - the Multimedia Message identifier is coded in hexadecimal.

## 8.84 Multimedia Message Transfer status

This clause applies if class "j" is supported.

Byte(s)	Description	Length
1	Multimedia Message Transfer Status tag	1
2	Length (X)	1
3 to 3+X	Multimedia Message Transfer Status	X

- Contents:
  - the Multimedia Message Transfer Status is response from the network to a multimedia message submission request;.
- Coding:
  - see "MM1\_submit.RES" message described in TS 123 140 [37].

## 8.85 MM Content Identifier

Byte(s)	Description	Length
1	MM Content Identifier tag	1
2	Length (X)	1
3 to X+2	MM Content Data Object tag	X

MM Content Data Object tag:

- Contents:
  - this contains the Data Object tag to be used when the MM Content is stored in the referenced BER-TLV file.
- Coding:
  - according to the different access technologies.

## 8.86 Multimedia Message Notification

Byte(s)	Description	Length
1	Multimedia Message Notification tag	1
2 to Y+2	Length (X)	1+Y
3+Y to X+(3+Y)	MMS notification message	X

- Contents:
  - the MMS notification message: "MM1\_notification.REQ" as specified in TS 123 140 [37].

## 8.87 Last Envelope

Byte(s)	Description	Length
1	Last Envelope tag	1
2	Length = 0	1

## 8.88 Registry application data

Byte(s)	Description	Length
1	Registry application data tag	1
2 to 1+Y	Length (3+X)	Y
2+Y to 3+Y	Application port number	2
4+Y	Data coding scheme	1
5+Y to 4+Y+X	Registry content	X

Application port number:

- Contents:
  - The application port number indicates on which TCP port the UICC can open a BIP channel in Terminal Server mode, to ask the handset to launch the application, and communicate with the application when running.
- Coding:
  - The application port number is coded in hexadecimal.

Data coding scheme is coded as for SMS Data coding scheme defined in TS 123 038 [3].

Registry content:

- Contents:
  - The Registry content contains the name and type of the application linked to the port number.
- Coding:
  - Byte 1:
    - '00': e-mail application;
    - '01': synchronization application other than '09';
    - '02': network monitoring application;
    - '03': video streaming application;
    - '04': audio streaming application;
    - '05': game application;
    - '06': browsing application;
    - '07': device management application as per OMA Device Management V1.2 specifications [41];
    - '08': device management application other than '07';
    - '09': data synchronization application as per OMA Data Synchronization V1.2.1 specifications [42];
    - '0A' to 'FE': RFU;
    - 'FF': unspecified type of application.
  - Byte 2 to X:
    - The name of the application is coded as indicated by the DCS.

## 8.89 Activate descriptor

This clause applies if class "I" is supported.

Byte(s)	Description	Length
1	Activate descriptor tag	1
2	Length	1
3	Target	1

Target:

- Coding:
  - '01' = UICC-CLF interface according to TS 102 613 [39];
  - '00' and '02' to 'FF' = RFU.

## 8.90 Broadcast Network information

Byte(s)	Description	Length
1	Broadcast Network Information tag	1
2	Length = X+1	1
3	Broadcast Network Technology	1
4 to 3+X	Broadcast Network Location Information	X

Broadcast Network Technology

- Contents:
  - The terminal shall use this information as a mechanism to indicate to the UICC the current Broadcast Network technology that it is using.
- Coding:
  - '00' = DVB-H;
  - '01' = DVB-T;
  - '02' = DVB-SH;
  - '03' = T-DMB;
  - '04' = FLO;
  - '05' = WiMAX;
  - All other values are reserved for future use.

Broadcast Network Location Information

- Contents:
  - The terminal shall use this information as a mechanism to indicate to the UICC the location information specific to the Broadcast Network technology that it is using.
- Coding:
  - Broadcast Network Location information coding is defined by the different Broadcast Network technologies.
  - For T-DMB, the coding is FFS.



- For DVB-H, DVB-T and DVB-SH, the Broadcast Network Location information is defined as follows:

Byte(s)	Description	Length
4 to 5	Network_id	2
6 to 7	Cell_id	2
8	Hierarchy	1
9	Number_of_subcell_id (Y)	1
10 to 9+Y	Subcell_id(s)	Y

"Network\_id" is defined in EN 300 468 [43] and is transmitted in the Network Information Table (NIT) according to EN 300 468 [43].

"Cell\_id" is defined in EN 300 468 [43] and for DVB-H is transmitted in the TPS bits of the DVB-H signal according to EN 302 304 [44].

"Hierarchy" defines the logical channel ("lp" for "low priority" or "hp" for "high priority") that is selected for reception when hierarchical modulation is used. For DVB-T, this field is not applicable. Coding of Hierarchy field is:

- '01'= low priority;
- '02'= high priority;
- 'FF'= not applicable.

"Subcell\_id", corresponds to cell\_id\_extension defined in EN 300 468 [43] and is transmitted as "cell\_id\_extension" in the Network Information Table (NIT) according to EN 300 468 [43].

- For FLO, the Broadcast Network Location information is defined as follows:

Byte(s)	Description	Length
4 to 5	Network_ID	2
6 to 7	WOI_ID	2
7 to 8	LOI_ID	2

"Network\_ID" is the identifier of the particular network defining the WOI\_ID and LOI\_ID, according to [45].

"WOI\_ID" is the Wide-Area Infrastructure ID, transmitted in Wide-Area OIS Channel, according to [45].

"LOI\_ID" is the Local-Area Infrastructure ID, transmitted in Wide-Area OIS Channel or RF Channel Description Message, according to [45].

- For WiMAX, the Broadcast Network Location information is defined as follows:

Byte(s)	Description	Length
4 to 6	NSP_ID	3
7 to 9	NAP_ID	3
10 to 15	BSID	6

"NSP\_ID" is the WiMAX Network Service Provider ID as specified in [46].

"NAP\_ID" is the WiMAX Network Access Provider ID as specified in [46].

"BSID" is the WiMAX Base Station Identifier as specified in [46].

## 8.91 Contactless state request

Byte(s)	Description	Length
1	Tag for contactless state request	1
2	Length	1
3	Contactless state request data	1

The coding of contactless state request data is as follows:

- '00' = Enable contactless; indicates that the handset requests to enable the contactless functionality in the UICC.
- '01' = Disable contactless; indicates that the handset requests to disable the contactless functionality in the UICC.
- '02' = Get contactless state; indicates that the handset requests the current state of the contactless functionality in the UICC.

## 8.92 Contactless functionality state

Byte(s)	Description	Length
1	Tag for contactless functionality state	1
2	Length	1
3	Contactless functionality state data	1

The coding of contactless interface state data is as follows:

- '00' = Enabled; indicates that the contactless functionality in the UICC is enabled.
- '01' = Disabled; indicates that the contactless functionality in the UICC is disabled.

## 8.93 Extended registry application data

Byte(s)	Description	Length
1	Extended registry application data tag	1
2 to 1+Y	Length (4+X)	Y
2+Y	Transport protocol type	1
3+Y to 4+Y	Application port number	2
5+Y	Data coding scheme	1
6+Y to 5+Y+X	Registry content	X

- Coding of the Transport protocol type:
  - '04': UDP, UICC in client mode, local connection;
  - '05': TCP, UICC in client mode, local connection;
  - '06': direct communication channel;
  - all other values are reserved;
  - UDP is defined in RFC 768 [9]; TCP is defined in RFC 793 [10].

Application port number:

- Contents:
  - The application port number indicates on which TCP or UDP port the UICC can open a BIP channel in Terminal Server mode, to ask the handset to launch the application, and communicate with the application when running, or the port number to be used as a reference for a direct communication channel.
- Coding:
  - The application port number is coded in hexadecimal.

Data coding scheme and registry content: As defined for registry application data.

## 8.94 eCAT client profile

Byte(s)	Description	Length
1	Tag for eCAT client profile	1
2 to Y+1	Length (X)	Y
Y+2 to X+Y+1	eCAT client profile data	X

The eCAT client profile data shall be coded as the command data of the TERMINAL PROFILE as specified in clause 5.2.

The bit for "CAT service list for eCAT client" indicates a request for the CAT service list for the eCAT client. If the eCAT client profile is sent in a secured eCAT session, the eCAT client shall set this bit and the UICC shall provide the CAT service list in its response.

## 8.95 eCAT client identity

Byte(s)	Description	Length
1	Tag for eCAT client identity	1
2	Length	1
3	eCAT client identifier	1

eCAT client identifier is the value specified in clause 8.7 (range '31' to '3F').

## 8.96 Encapsulated envelope type

Byte(s)	Description	Length
1	Tag for encapsulated envelope type	1
2	Length	1
3	Encapsulated envelope type data	0 or 1

Encapsulated envelope type data contains the BER TLV tag of the encapsulated envelope.

Length equal to zero indicates the end of an encapsulated command session.

## 8.97 Call control result

Byte(s)	Description	Length
1	Tag for Call control result	1
2	Length	1
3	Call control result	1

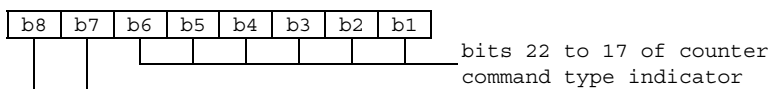
Call control result:

- Contents: the command that the UICC gives to the terminal concerning whether to allow, bar or modify the proposed call;
- Coding:
  - '00' = Allowed, no modification;
  - '01' = Not allowed;
  - '02' = Allowed with modifications.

## 8.98 eCAT sequence number

Byte(s)	Description	Length
1	Tag for eCAT sequence number	1
2	Length = 3	1
3	Command type indicator and highest 6 bits of counter	1
4 to 5	lower 16 bits of counter	2

Coding of byte 3:



Command type indicator values:

- 00: command container;
- 01: terminal response;
- 10: envelope/profile container event;
- 11: envelope/profile container event response.

The UICC and the eCAT client shall manage two separate counters, one for commands and one for envelopes/profile container events. The counters shall be set to zero upon setting up the security for the connection to the eCAT client. The UICC shall increment the command counter before issuing a secured encapsulated command. The eCAT client shall increment the envelope/profile container event counter before issuing a secured encapsulated envelope or a secured profile container event. Terminal and envelope/profile container event responses shall carry the same counter value as the corresponding command resp. envelope/event.

## 8.99 Encrypted TLV list

Byte(s)	Description	Length
1	Tag for Encrypted TLV list	1
2	Length	1 or 2
3	Encrypted TLV list data	N

For an encrypted Encapsulated Profile, the encrypted TLV list data contains the comprehension TLVs of the eCAT sequence number, the text string and the eCAT client profile, padded and encrypted as defined for the encryption method used with the eCAT client.

For a command, the encrypted TLV list data contains the comprehension TLVs of the encapsulated command, starting from eCAT sequence number, padded and encrypted as defined for the encryption method used with the eCAT client.

For a terminal response, the encrypted TLV list data contains the comprehension TLVs of the encapsulated terminal response, starting from eCAT sequence number, padded and encrypted as defined for the encryption method used with the eCAT client.

For an envelope and an envelope response, the encrypted TLV list data contains the comprehension TLVs of the encapsulated envelope respectively the encapsulated envelope response, as it would be provided for a non-encrypted encapsulated envelope respectively encapsulated envelope response, i.e. starting with the eCAT sequence number, padded and encrypted as defined for the encryption method used with the eCAT client.

## 8.100 MAC

Byte(s)	Description	Length
1	Tag for MAC	1
2	Length	1 or 2
3	MAC data	N

For a non-encrypted and an encrypted Encapsulated Profile, the MAC data contains the MAC calculated across the comprehension TLVs of the Encapsulated Profile, starting from the event list, as defined for the MACing method used with the eCAT client.

For the encapsulated profile response, the MAC data contains the MAC calculated across the comprehension TLV of the eCAT client identity, as defined for the MACing method used with the eCAT client.

For a non-encrypted encapsulated command or terminal response, the MAC data contains the MAC calculated across the comprehension TLVs of the encapsulated command, starting from the eCAT sequence number, as defined for the MACing method used with the eCAT client.

For a non-encrypted encapsulated envelope or envelope response, the MAC data contains the MAC calculated across the comprehension TLVs of the encapsulated envelope resp. envelope response, starting from the eCAT sequence number, as defined for the MACing method used with the eCAT client.

For an encrypted encapsulated command, terminal response, envelope or envelope response, the MAC data contains the MAC calculated across the Encrypted TLV list data object as defined for the MACing method used with the eCAT client.

## 8.101 SA template

Byte(s)	Description	Length
1	Tag for SA template	1
2	Length	1 or 2
3	SA template data	N

The content and coding of the SA template data is defined as follows:

### For the request Master SA setup command:

The SA template data shall contain the following TLV object defined in TS 102 221 [1] for the MANAGE SECURE CHANNEL - Establish SA - Master SA command:

- Term label - UICC\_appli\_ID

This object shall be used by the terminal in its terminal response.

### For the terminal response to a request Master SA setup:

The SA template data shall contain the following TLV objects defined in TS 102 221 [1] for the MANAGE SECURE CHANNEL - Establish SA - Master SA command:

- Key Agreement Mechanism.
- Term label - Terminal\_ID.
- Term label - Terminal\_appli\_ID.
- Term label - UICC\_Identifier.
- Term label - UICC\_appli\_ID.

Only proprietary pre agreed strong preshared keys are supported for eCAT.

**For the request Connection SA setup command:**

The SA template data shall contain the following TLV objects defined in TS 102 221 [1] for the MANAGE SECURE CHANNEL - Establish SA - Master SA response:

- Key Agreement Mechanism.
- MSA\_ID.

**For the terminal response to a request Connection SA setup:**

The SA template data shall contain the following TLV objects defined in TS 102 221 [1] for the MANAGE SECURE CHANNEL - Establish SA - Connection SA command:

- Algorithm and integrity.
- MSA\_ID.
- Tnonce.

**For the request Secure Channel Start command:**

The SA template data shall contain the following TLV objects defined in TS 102 221 [1] for the MANAGE SECURE CHANNEL - Establish SA - Connection SA response:

- Algorithm and integrity.
- CSA\_ID.
- Unonce.
- CSAMAC.

**For the terminal response to a request Secure Channel Start:**

The SA template data shall contain the following TLV objects defined in TS 102 221 [1] for the MANAGE SECURE CHANNEL - Establish SA - Start Secure Channel command:

- Algorithm and integrity.
- CSA\_ID.
- SSCMAC.
- Endpoint data container size.

Endpoint data container size shall be empty as it is not used for eCAT.

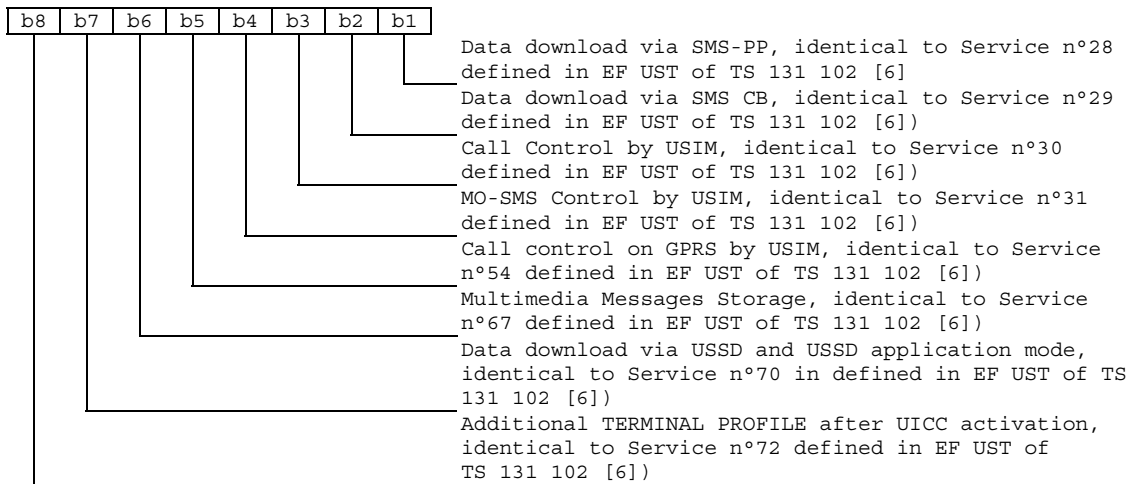
## 8.102 CAT service list

Byte(s)	Description	Length
1	Tag for CAT service list	1
2	Length (X)	1
3 to X+2	CAT service list data	X

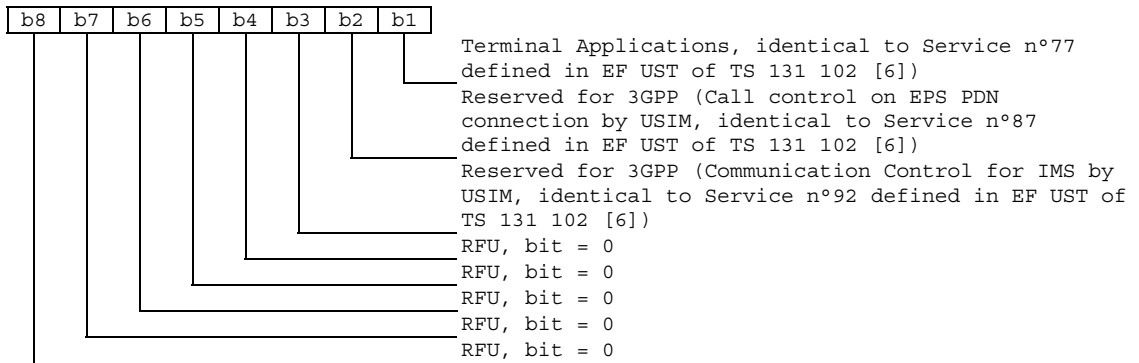
The CAT service list data shall indicate the services available for the terminal in a specific NAA or the services available for an eCAT client and is coded as defined below.

Data provided if the CAT service list is requested for a USIM as defined in TS 131 102 [6]:

First byte:

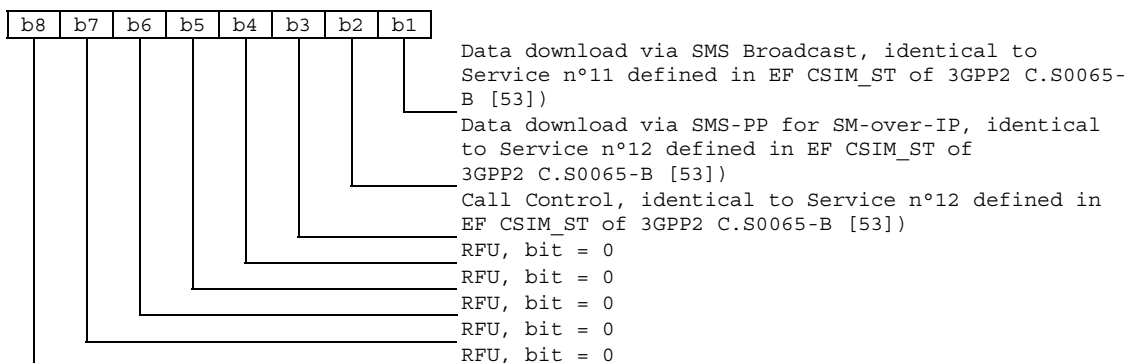


Second byte:



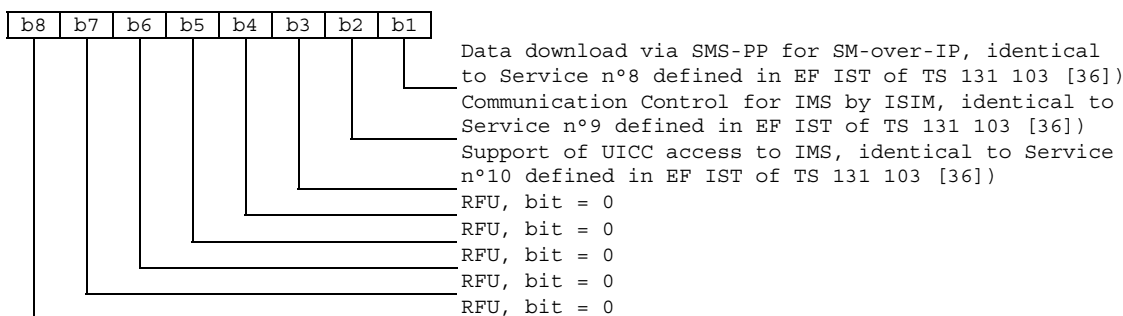
Data provided if the CAT service list is requested for a CSIM as defined in 3GPP2 C.S0065-B [53]:

First byte:



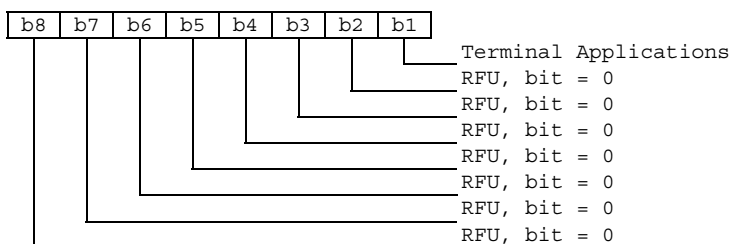
Data provided if the CAT service list is requested for an ISIM as defined in 3GPP2 TS 131 103 [36]:

First byte:



Data provided if the CAT service list is requested for an eCAT client:

First byte:



For all variants, additional bytes may be added in the future.

The bits set by the UICC in the CAT service list for an NAA shall be identical to the values set in the respective service tables of the NAA.

## 9 Tag values

This clause specifies the tag values used to identify the BER-TLV and COMPREHENSION-TLV data objects used in the present document, and reserves the technology specific Tags.

### 9.1 BER-TLV tags in terminal to UICC direction

See TS 101 220 [31].



## 9.2 BER-TLV tags in UICC to terminal direction

Description	Length of tag	Value
Proactive UICC command tag	1	'D0'

## 9.3 COMPREHENSION-TLV tags in both directions

See TS 101 220 [31].

## 9.4 Type of command and next action indicator

The table below shows the values which shall be used for Type of Command coding (see clause 8.6) and Next Action Indicator coding (see clause 8.24).

Value	Name	Used for Type of Command coding	Used for Next Action Indicator coding
'00'		-	-
'01'	REFRESH	X	
'02'	MORE TIME	X	
'03'	POLL INTERVAL	X	
'04'	POLLING OFF	X	
'05'	SET UP EVENT LIST	X	
'10'	SET UP CALL	X	X
'11'	Reserved for GSM/3G (SEND SS)	X	X
'12'	Reserved for GSM/3G (SEND USSD)	X	X
'13'	SEND SHORT MESSAGE	X	X
'14'	SEND DTMF	X	
'15'	LAUNCH BROWSER	X	X
'16'	Reserved for 3GPP (GEOGRAPHICAL LOCATION REQUEST)	X	
'20'	PLAY TONE	X	X
'21'	DISPLAY TEXT	X	X
'22'	GET INKEY	X	X
'23'	GET INPUT	X	X
'24'	SELECT ITEM	X	X
'25'	SET UP MENU	X	X
'26'	PROVIDE LOCAL INFORMATION	X	
'27'	TIMER MANAGEMENT	X	
'28'	SET UP IDLE MODE TEXT	X	X
'30'	PERFORM CARD APDU	X	X
'31'	POWER ON CARD	X	X
'32'	POWER OFF CARD	X	X
'33'	GET READER STATUS	X	X
'34'	RUN AT COMMAND	X	
'35'	LANGUAGE NOTIFICATION	X	
'40'	OPEN CHANNEL	X	X
'41'	CLOSE CHANNEL	X	X
'42'	RECEIVE DATA	X	X
'43'	SEND DATA	X	X
'44'	GET CHANNEL STATUS	X	X
'45'	SERVICE SEARCH	X	X
'46'	GET SERVICE INFORMATION	X	X
'47'	DECLARE SERVICE	X	
'50'	SET FRAMES	X	
'51'	GET FRAMES STATUS	X	
'60'	(RETRIEVE MULTIMEDIA MESSAGE)	X	X
'61'	(SUBMIT MULTIMEDIA MESSAGE)	X	X
'62'	DISPLAY MULTIMEDIA MESSAGE	X	X
'70'	ACTIVATE	X	
'71'	CONTACTLESS STATE CHANGED	X	
'72'	COMMAND CONTAINER	X	

Value	Name	Used for Type of Command coding	Used for Next Action Indicator coding
'73'	ENCAPSULATED SESSION CONTROL	X	
'81'	End of the proactive session	not applicable	X
'F0' to 'FE'	Reserved for proprietary use	X	X

## 10 Allowed type of command and device identity combinations

Only certain types of commands can be issued with certain device identities. These are defined below.

Command description	Source	Destination
CALL CONTROL	Terminal	UICC
COMMAND RESULT	Terminal	UICC
DISPLAY TEXT	UICC	Display
EVENT DOWNLOAD		
- MT call	Network	UICC
- Call connected at near end (MT call)	Terminal	UICC
- Call connected at far end (MO call)	Network	UICC
- Call disconnected at near end	Terminal	UICC
- Call disconnected at far end	Network	UICC
- Location status	Terminal	UICC
- User activity	Terminal	UICC
- Idle screen available	Display	UICC
- Card reader status	Terminal	UICC
- Language selection	Terminal	UICC
- Browser termination	Terminal	UICC
- Data available	Terminal	UICC
- Channel status	Terminal	UICC
- Access Technology Change	Terminal	UICC
- Display parameters changed	Terminal	UICC
- Local connection	Network	UICC
- Network Search Mode Change	Terminal	UICC
- Browsing status	Terminal	UICC
- Frames Information changed	Terminal	UICC
- HCI Connectivity	Terminal	UICC
- Contactless state request	Terminal	UICC
- Profile Container	Terminal or eCAT client x	UICC
GET INKEY	UICC	Terminal
GET INPUT	UICC	Terminal
GET READER STATUS		
- If card reader status requested	UICC	Terminal
- If card reader identifier requested	UICC	Card reader x
LANGUAGE NOTIFICATION	UICC	Terminal
LAUNCH BROWSER	UICC	Terminal
MENU SELECTION	Keypad	UICC
MORE TIME	UICC	Terminal
PERFORM CARD APDU	UICC	Card reader x
PLAY TONE	UICC	Earpiece (see note)
POLLING OFF	UICC	Terminal
POLL INTERVAL	UICC	Terminal
POWER ON CARD	UICC	Card reader x
POWER OFF CARD	UICC	Card reader x
PROFILE DOWNLOAD	Terminal	UICC
PROVIDE LOCAL INFORMATION	UICC	Terminal
REFRESH	UICC	Terminal
RUN AT COMMAND	UICC	Terminal
SELECT ITEM	UICC	Terminal
SEND DTMF	UICC	Network
SEND SHORT MESSAGE	UICC	Network
SET UP CALL	UICC	Network

Command description	Source	Destination
SET UP EVENT LIST	UICC	Terminal
SET UP IDLE MODE TEXT	UICC	Terminal
SET UP MENU	UICC	Terminal
TIMER MANAGEMENT	UICC	Terminal
TIMER EXPIRATION	Terminal	UICC
OPEN CHANNEL	UICC	Terminal
CLOSE CHANNEL	UICC	Channel x
RECEIVE DATA	UICC	Channel x
SEND DATA	UICC	Channel x
GET CHANNEL STATUS	UICC	Terminal
SERVICE SEARCH	UICC	Terminal
GET SERVICE INFORMATION	UICC	Terminal
DECLARE SERVICE	UICC	Terminal
SET FRAMES	UICC	Terminal
GET FRAMES STATUS	UICC	Terminal
RETRIEVE MULTIMEDIA MESSAGE	UICC	Network
SUBMIT MULTIMEDIA MESSAGE	UICC	Network
MMS Transfer Status	Network	UICC
DISPLAY MULTIMEDIA MESSAGE	UICC	Terminal
MMS notification download	Network	UICC
TERMINAL APPLICATION	Terminal	UICC
ACTIVATE	UICC	Terminal
ENVELOPE (ENVELOPE CONTAINER)	eCAT client x	UICC
COMMAND CONTAINER	UICC	eCAT client x
ENCAPSULATED SESSION CONTROL	UICC	eCAT client x
ENVELOPE (SERVICE LIST)	Terminal	UICC

NOTE: The terminal may route the tone to other loudspeakers (external ringer, car kit) if more appropriate.

---

## 11 Void

## Annex A (normative): Support of CAT by terminal equipment

Support of CAT is optional for terminal Equipment. However, if a terminal states conformance with a specific CAT release, it is mandatory for the terminal to support all functions of that release.

The support of letter classes, which specify mainly terminal hardware dependent features, is optional for the terminal and may supplement the CAT functionality described in the present document. If a terminal states conformance to a letter class, it is mandatory to support all functions within the respective letter class.

Table A.1 indicates the commands and functions of the optional letter classes.

**Table A.1: Description of letter classes**

Letter classes	Command/function description
a	Proactive command: GET READER STATUS Proactive command: PERFORM CARD APDU Proactive command: POWER ON CARD Proactive command: POWER OFF CARD Event download: Card reader status
b	Proactive command: RUN AT COMMAND
c	Proactive command: LAUNCH BROWSER Event download: Browser termination event Event download: Browsing status event
d	Soft key support
e	Proactive command: OPEN CHANNEL Proactive command: CLOSE CHANNEL Proactive command: RECEIVE DATA Proactive command: SEND DATA Proactive command: GET CHANNEL STATUS Event download: Data available Event download: Channel status
f	Proactive command: SERVICE SEARCH Proactive command: GET SERVICE INFORMATION Proactive command: DECLARE SERVICE Event download: Local connection event
g	Proactive Command: PROVIDE LOCAL INFORMATION (Battery State)
h	Multi-media Call support
i	Proactive command: SET FRAMES Proactive command: GET FRAMES STATUS Event download: Frames Information changed
j	Proactive command: RETRIEVE MULTIMEDIA MESSAGE
	Proactive command: SUBMIT MULTIMEDIA MESSAGE
	Proactive command: DISPLAY MULTIMEDIA MESSAGE
	Envelope command: MMS notification download
k	Envelope command: MMS Transfer status
	Envelope command: TERMINAL APPLICATIONS
l	Proactive command: ACTIVATE
m	Event download: HCI connectivity event
n	See TS 131 111 [26] (Proactive command: Geographical Location Request and Envelope command: Geographical Location Reporting)
o	Proactive Command: PROVIDE LOCAL INFORMATION (Broadcast Network Information)
p	See TS 131 111 [26] (USSD Data download in application mode)
q	See TS 131 111 [26] (Proactive command: Provide Local Information (CSG cell discovery) and Event download: CSG cell selection)

<b>Letter classes</b>	<b>Command/function description</b>
r	Proactive command: CONTACTLESS STATE CHANGED Event download: Contactless state request
s	Support of CAT over the modem interface
t	See TS 131 111 [26] (UICC access to IMS)
u	Event download: Profile Container, Envelope command: ENVELOPE CONTAINER, Proactive Commands: COMMAND CONTAINER and ENCAPSULATED SESSION CONTROL
v	See TS 131 111 [26] (Provide Local Information (H(e)NB IP address))
w	See TS 131 111 [26] (Provide Local Information (H(e)NB surrounding macrocells))
x	Security for Profile Container, Envelope Container, COMMAND CONTAINER and ENCAPSULATED SESSION CONTROL
y	Envelope command: Service List Retrieval

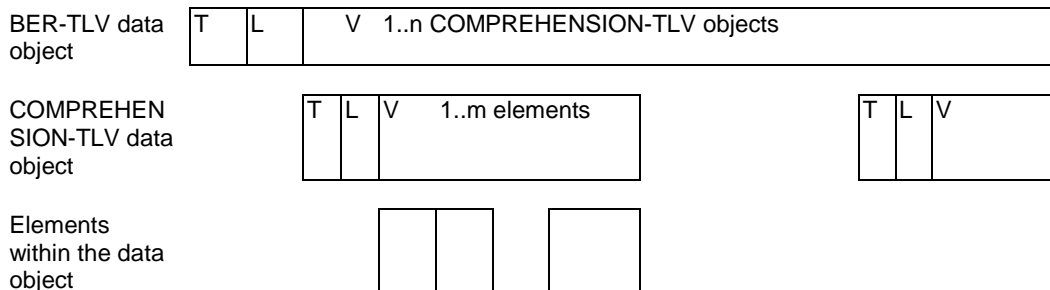
## Annex B (informative): Example of DISPLAY TEXT proactive UICC command

Example of DISPLAY TEXT Proactive UICC Command (BER-TLV Data Object).

**Table B.1: Example of DISPLAY TEXT**

Byte#	Value (Hex)	Description
1	D0	Proactive UICC command tag
2	15	Length
3	81	Command details tag
4	03	Length
5	01	Command number
6 to 7	21 00	Display text (normal priority, clear message after a delay)
8	82	Device identities tag
9	02	Length
10	81	Source: UICC
11	02	Destination: Display
12	8D	Text string tag
13	04	Length
14	04	Data coding scheme ('04'=8-bit default SMS)
15 to 17	43, 41, 54	Text string ("CAT")
18	C8	Text attribute tag
19	04	Length
20 to 23	01, 02, 03, 04	Text Formatting

# Annex C (normative): Structure of CAT communications



**Figure C.1**

CAT commands and responses are sent across the interface as BER-TLV data objects. Each APDU shall only contain one BER-TLV object. See TS 101 220 [31] for more information on data objects.

The tag of a BER-TLV is a constant value, length one byte, indicating it is a CAT command.

The length is coded onto 1 or 2 bytes according to TS 101 220 [31]. Table C.1 details this coding.

**Table C.1**

Length	Byte 1	Byte 2
0-127	length ('00' to '7F')	not present
128-255	'81'	length ('80' to 'FF')

Any length within the APDU limits (up to 255 bytes) can thus be encoded on two bytes. This coding is chosen to remain compatible with TS 101 220 [31].

Any values for byte 1 or byte 2 that are not shown above shall be treated as an error and the whole message shall be rejected.

The value part of the BER-TLV data object consists of COMPREHENSION-TLV data objects, as shown in the description of the COMPREHENSION-TLV data objects on individual commands. It is mandatory for COMPREHENSION-TLV data objects to be provided in the order given in the description of each command. New COMPREHENSION-TLV data objects can be added to the end of a command.

The structure of COMPREHENSION-TLV tags is defined in TS 101 220 [31].

The M/O/C columns specify whether it is mandatory, optional or conditional for the sender to send that particular COMPREHENSION-TLV data object for compliance with the current version of the present document. The Min (Minimum Set) column describes whether it is necessary for the receiver to have received that particular COMPREHENSION-TLV data object to be able to attempt at least the most basic form of this command. The procedure for dealing with incomplete messages is described in clause 6.10.

'00' and 'FF' are never used as tag values. This is in accordance with TS 101 220 [31]. Padding characters are not allowed.

# Annex D (informative): Terminal display in proactive UICC session

Example of the terminal display whilst the terminal is in a proactive UICC session.

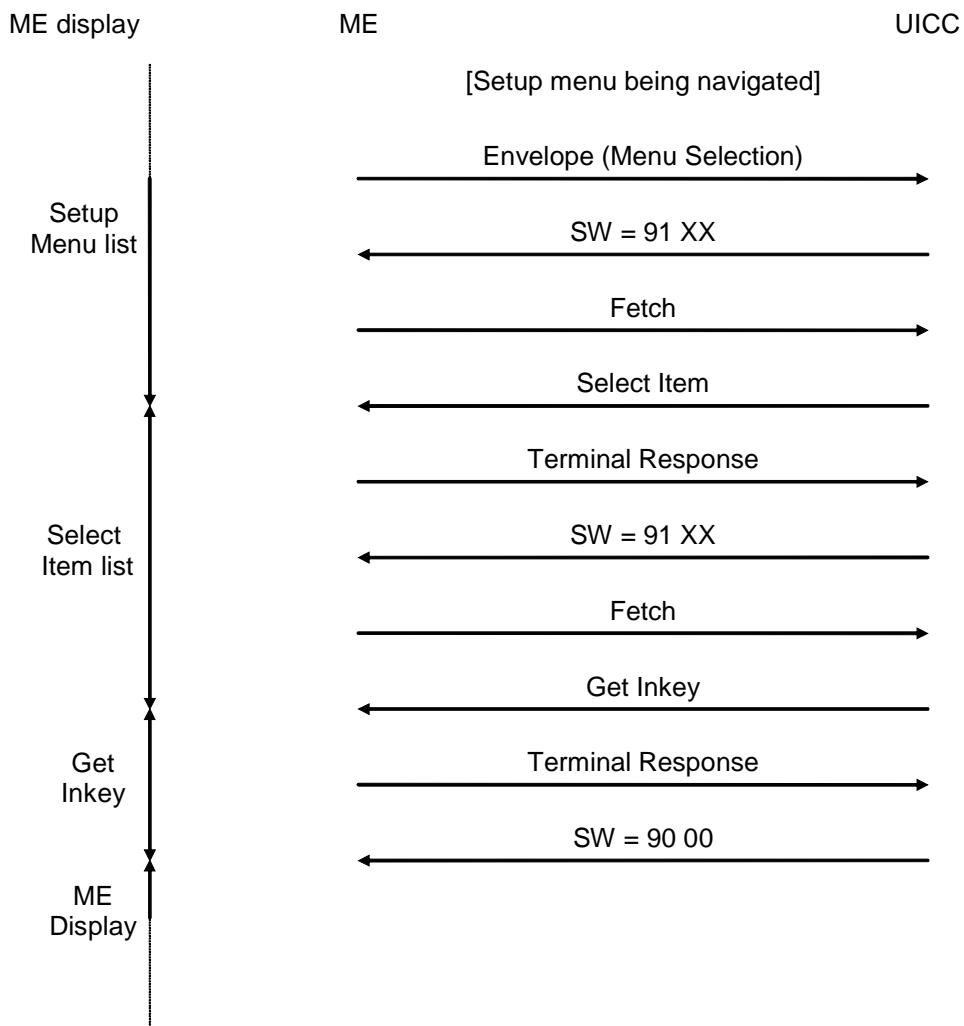


Figure D.1



# Annex E (informative): Help information feature processing

The following example shows the use of the commands Menu Selection/Select Item and Get Input in conjunction with the help information feature.

Terminal	UICC
TERMINAL PROFILE	-----> <----- 91xx
FETCH	-----> <----- SET UP MENU (Help available)
TERMINAL RESPONSE (OK)	-----> <----- 90 00
...	
ENVELOPE (MENU SELECTION, help on menu item m)	-----> <----- 91 xx
FETCH	-----> <----- DISPLAY TEXT (Help info to item m)
TERMINAL RESPONSE (OK)	-----> <----- 90 00
(Terminal offers menu again and user selects item m)	
ENVELOPE (MENU SELECTION, select item m)	-----> <----- 91 xx
FETCH	-----> <----- SELECT ITEM (Item list under item m, help available)
TERMINAL RESPONSE (Help on item mn in item list under item m)	-----> <----- 91 xx
FETCH	-----> <----- DISPLAY TEXT (Help info to item mn)
TERMINAL RESPONSE (OK)	-----> <----- 91 xx
FETCH	-----> <----- Repetition of SELECT ITEM (Item list under item m, help available)
...	
FETCH	-----> <----- 91xx
FETCH	-----> <----- GET INPUT
TERMINAL RESPONSE (Help info required)	-----> <----- 91 xx
FETCH	-----> <----- DISPLAY TEXT (Help info)
TERMINAL RESPONSE (OK)	-----> <----- 91 xx
FETCH	-----> <----- Repetition of GET INPUT
TERMINAL RESPONSE (OK)	-----> ...
...	

Figure E.1

## Annex F (informative): Monitoring of events

Some of the events monitored through the event download mechanism are reported by the terminal each time the event occurs, while other events are reported only once (the terminal removes the event type from the current event list once the event occurs). This is summarized in table F.1.

**Table F.1**

<b>Event</b>	<b>Continuously reported</b>	<b>Reported once</b>
MT call	X	
Call connected	X	
Call disconnected	X	
Location status	X	
User activity		X
Idle screen available		X
Card reader status	X	
Language selection	X	
Data available	X	
Channel status	X	
Browser termination	X	
Access Technology Change	X	
Display parameters changed	X	
Local connection	X	
Network Search Mode Change	X	
Browsing status	X	
Frames Information changed	X	
HCI Connectivity	X	
Contactless state request	X	

---

## Annex G (normative): Support of multiple card operation

This annex applies if class "a" is supported.

It is intended that Multiple Card commands are an optional extension to the basic CAT functionality in the present document.

The terminal is responsible for appropriate protocol management, as defined in ISO/IEC 7816-4 [14]. This includes APDU mapping and procedure byte handling.

If the terminal is already powered on and a UICC is active, then, when Card x is inserted, the terminal powers on Card x. The terminal shall identify if Card x contains the UICC application. If it does, TS 100 922 [19] applies. If it does not contain the UICC application, or it is not selected by the user for 3G operation, then the terminal powers off Card x. If applicable, the terminal shall send an event download (card reader status) message to the current UICC. When required, the CAT application of the current UICC card shall power on Card x and control communications, through the relevant proactive commands.

When the terminal is powered on, the terminal locates and selects the preferred UICC card defined in TS 100 922 [19]. If applicable, the terminal sends a terminal Profile command to the UICC. When required, the CAT application issues a Get Reader Status proactive command, which gets information on all readers and cards available to the CAT application. This procedure also applies if the terminal is already powered on with no UICC present, and a card is then inserted.

When the UICC issues a POWER ON CARD, and the terminal successfully receives an Answer To Reset from Card x, the terminal shall return a successful terminal Response containing the ATR, even if it does not understand the contents of the ATR, or support any of the protocols indicated.

The terminal shall ensure that Card x is deactivated according to ISO/IEC 7816-3 [13]. Where deactivation is not due to a POWER OFF CARD proactive command (e.g. card removed, card reader removed, or low battery), the event download (card reader status) procedure may also be applicable.

# Annex H (informative): Multiple card proactive command examples

This annex applies if class "a" is supported.

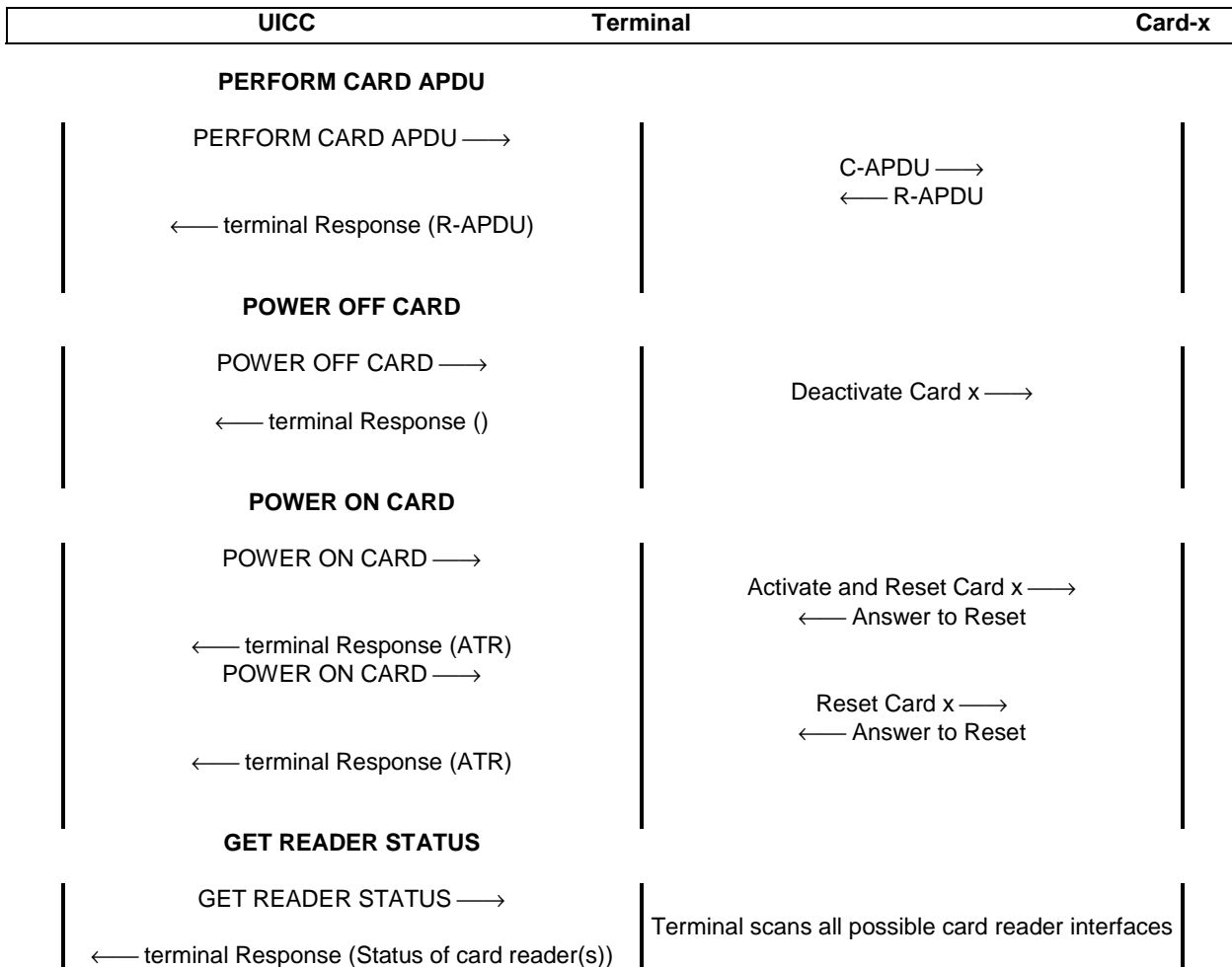
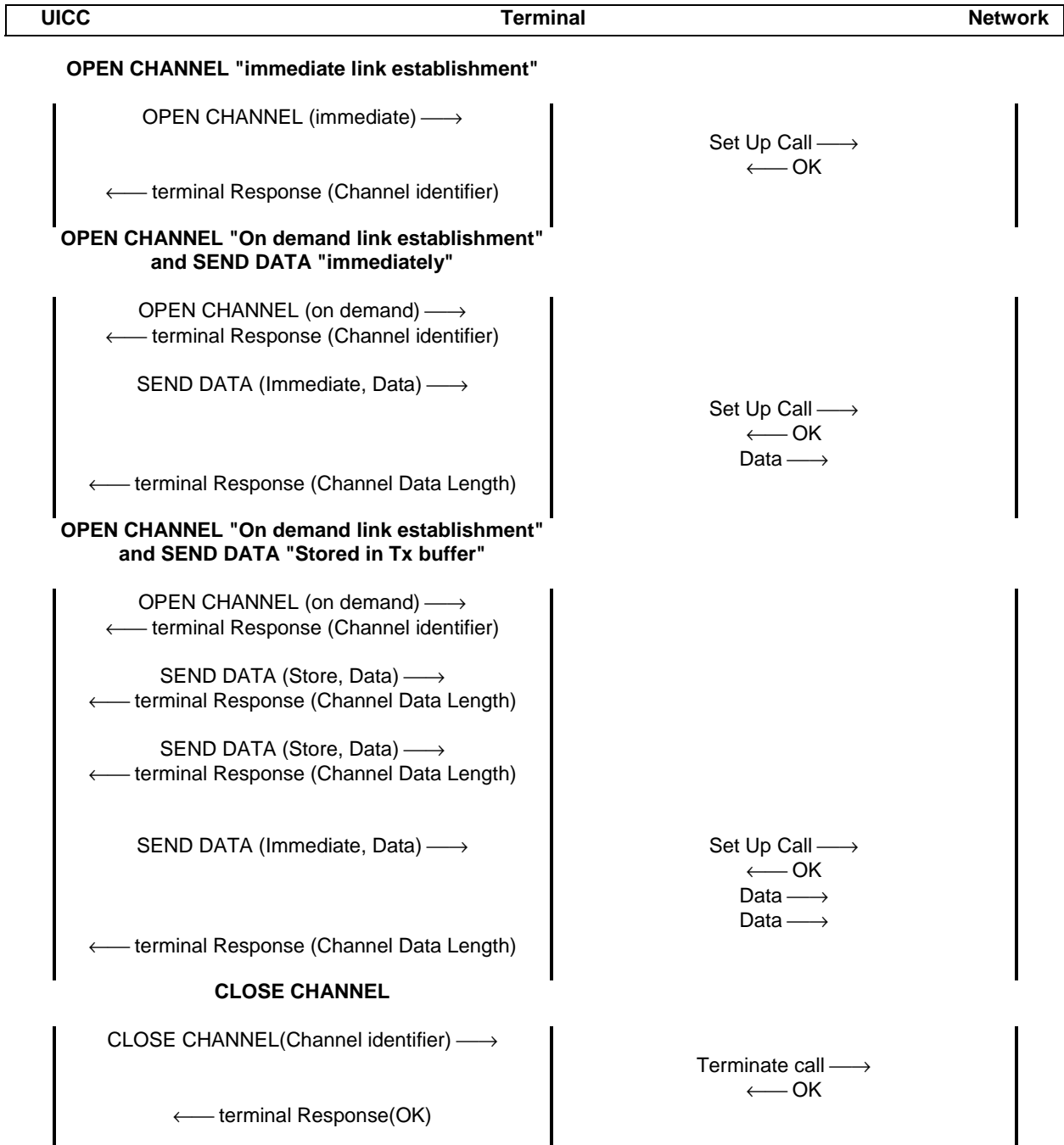


Figure H.1

# Annex I (informative): Bearer independent protocol proactive command examples

This annex applies if class "e" is supported.



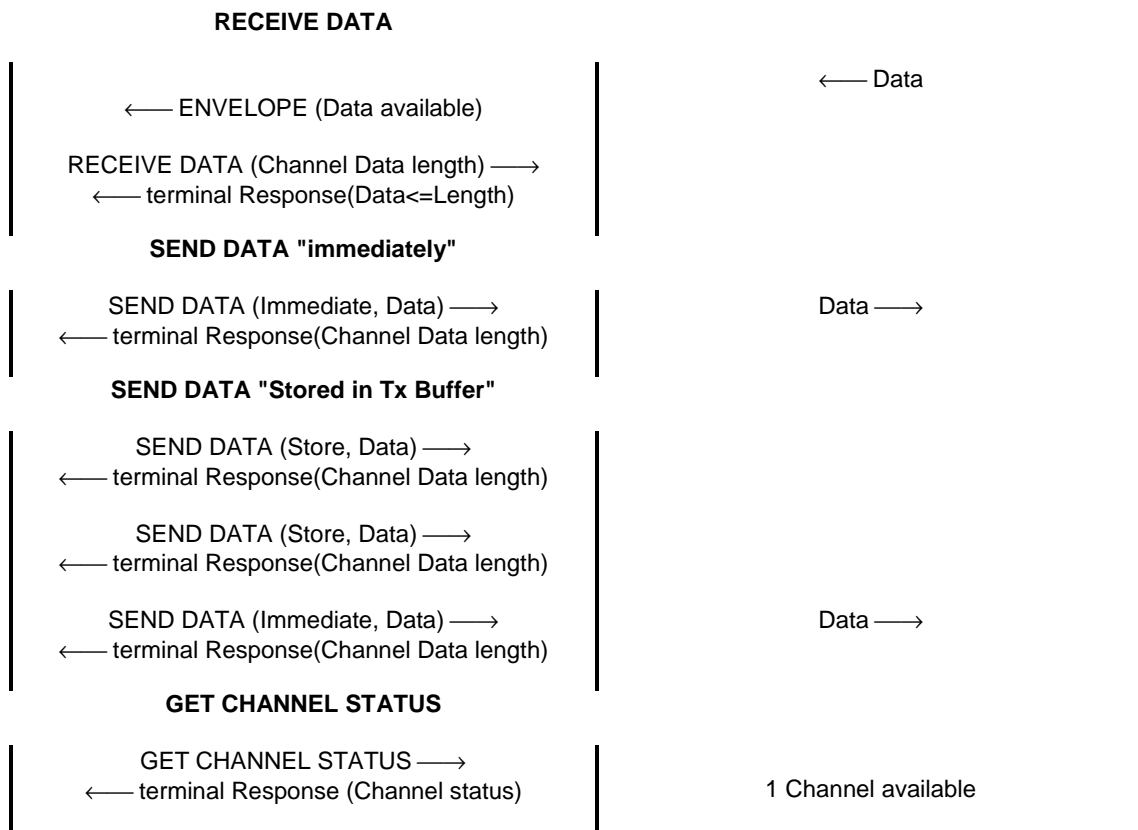
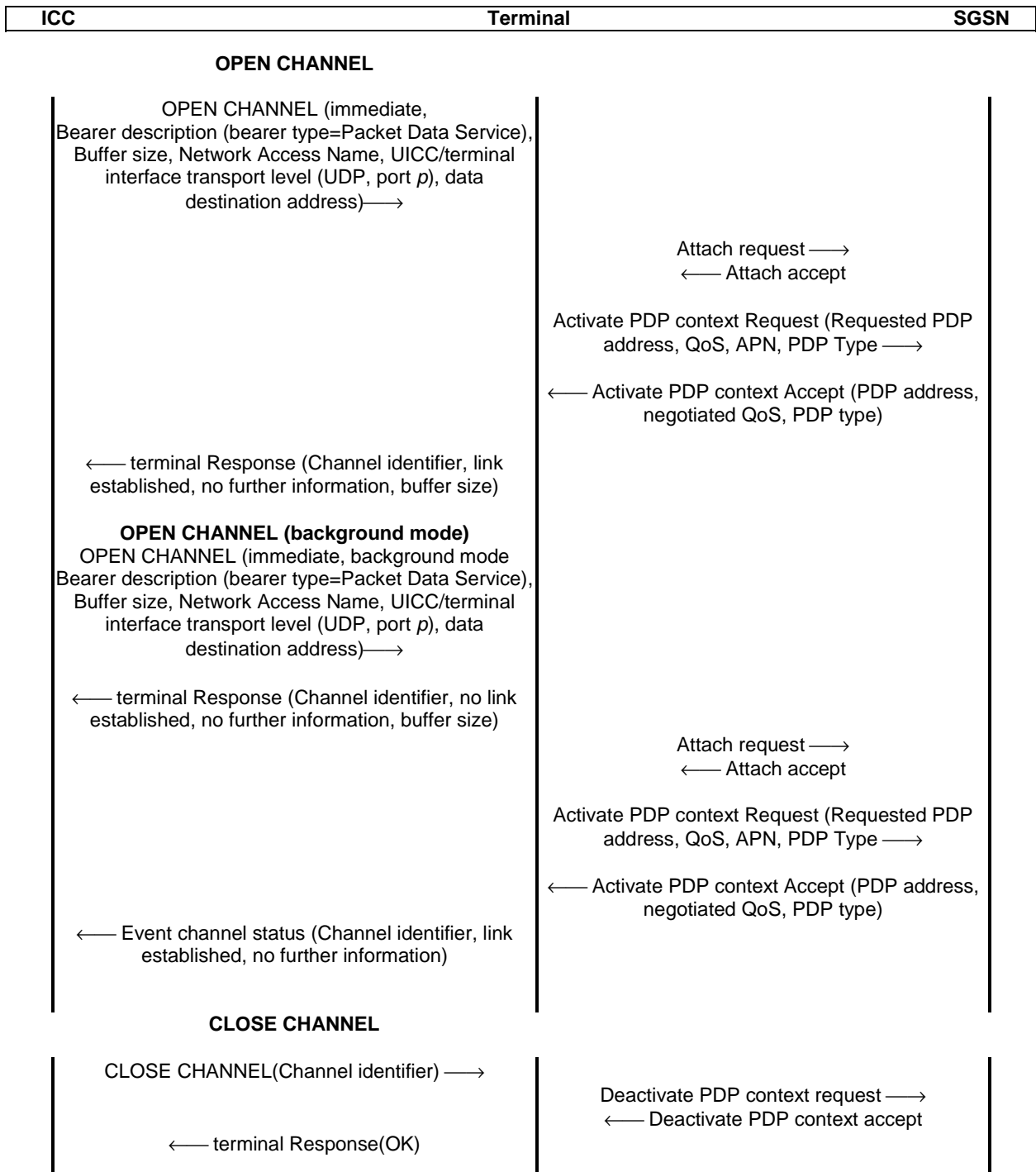
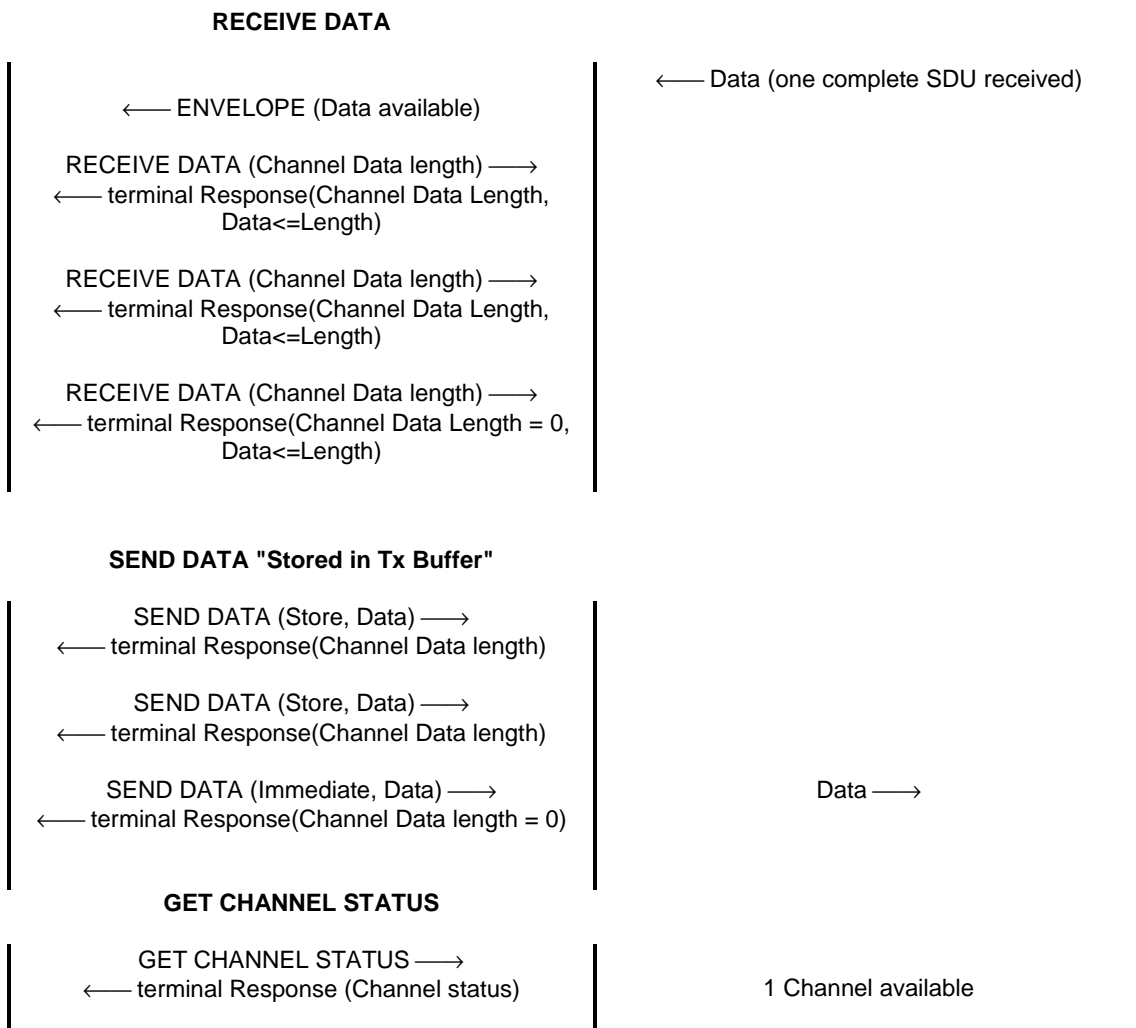


Figure I.1

Example for Packet Data Service bearer applied to GPRS:





**Figure I.2**



Example for OPEN CHANNEL in Terminal Server Mode, where the UICC provides additional launch parameters. This example applies if class "e" and class "k" are supported.

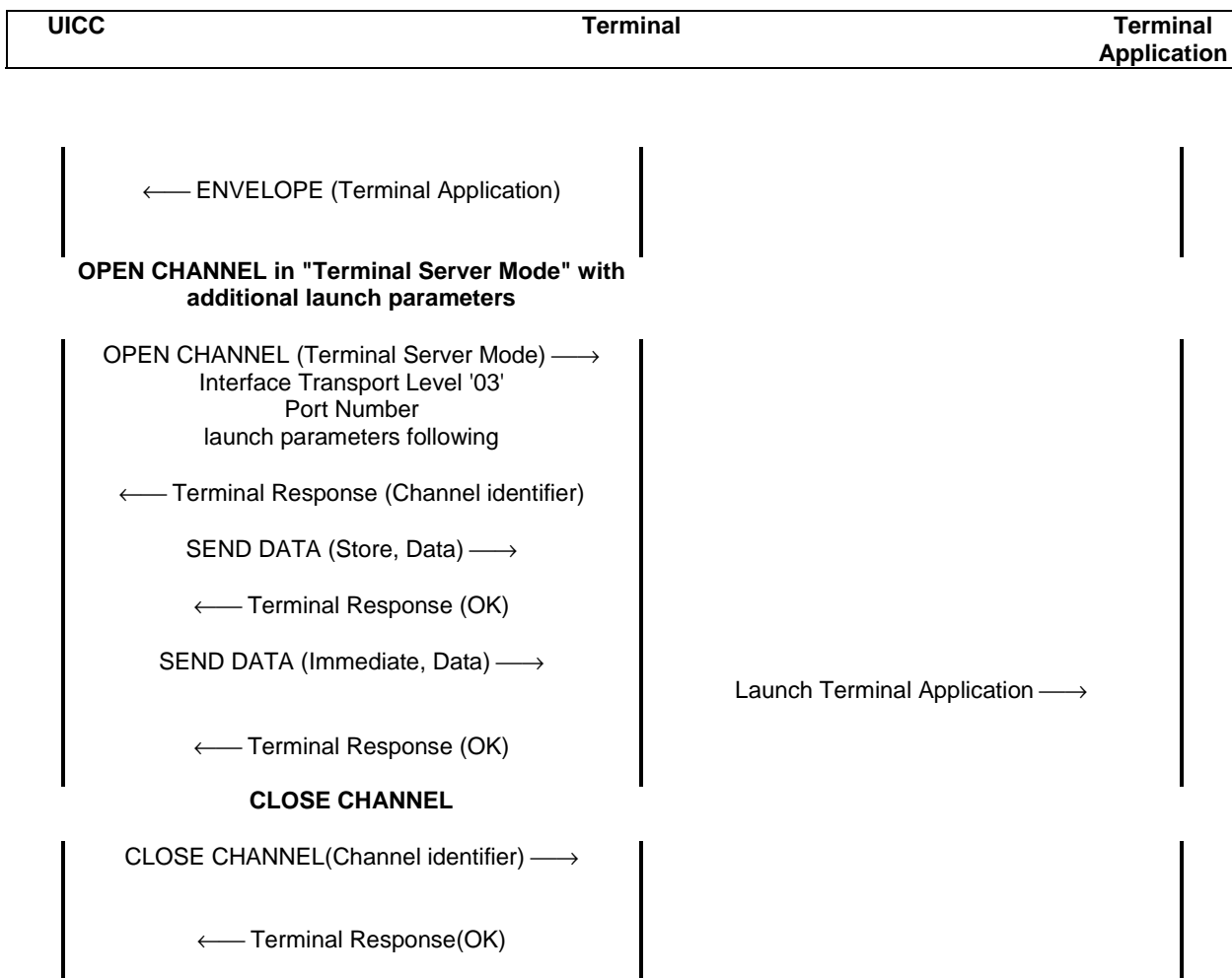


Figure I.3

---

## Annex J (informative): WAP terminology

### References:

- WAP specifications: <http://www.wapforum.org/>.
- WAP Smart card provisioning specification: <http://www.wapforum.org/>.

### Definitions:

**WAE User Agent:** any software or device that interprets WML, WMLScript

**WMLScript:** scripting language used to run a program in the terminal device

### Abbreviations:

WAE	Wireless Application Environment
WAP	Wireless Application Protocol
WML	Wireless Markup Language

---

## Annex K (informative): Use of CAT bearer independent protocol for local links Bluetooth case

Bluetooth services to be run by the UICC should be developed so that the access to their service record is open and does not necessitate any security mechanism (no authentication or encryption).

---

### K.1 Service search command

The Local Bearer Technology Identifier is Bluetooth. Service Search consists for the terminal in first performing a device discovery of the devices that conform to the Device Filter (inquiry responses are filtered according to the list of Class of Device given in the Device Filter); then performing an SDP\_ServiceSearchRequest, as defined in the Bluetooth specification [16], on each device to check the support of the given service. The terminal shall then return the Service Availability data object which is a list of BD\_ADDR, ServiceRecordHandle, CoD and Device Name.

Note for Handset Manufacturers:

- as the mobile is not always connected to other devices present in the remote environment (e.g. Bluetooth), when performing a service search, it is up to the terminal to set a procedure that allows:
  - a "scan" of the environment to discover new devices;
  - a connection to Service Discovery Servers of discovered devices;
  - a match with the requested service to set up the response to the CAT application.

---

### K.2 Get service information command

The Local Bearer Technology Identifier is Bluetooth, Get Service Information consists for the terminal in connecting to a specific device and performing an SDP\_ServiceAttributeRequest PDU as defined in the Bluetooth specification [16]. The terminal shall then return the Service Record data object.

NOTE: When performing a GET SERVICE INFORMATION, it is up to the terminal to set up a connection with the requested device and perform the SDP exchange.

---

### K.3 OPEN CHANNEL command

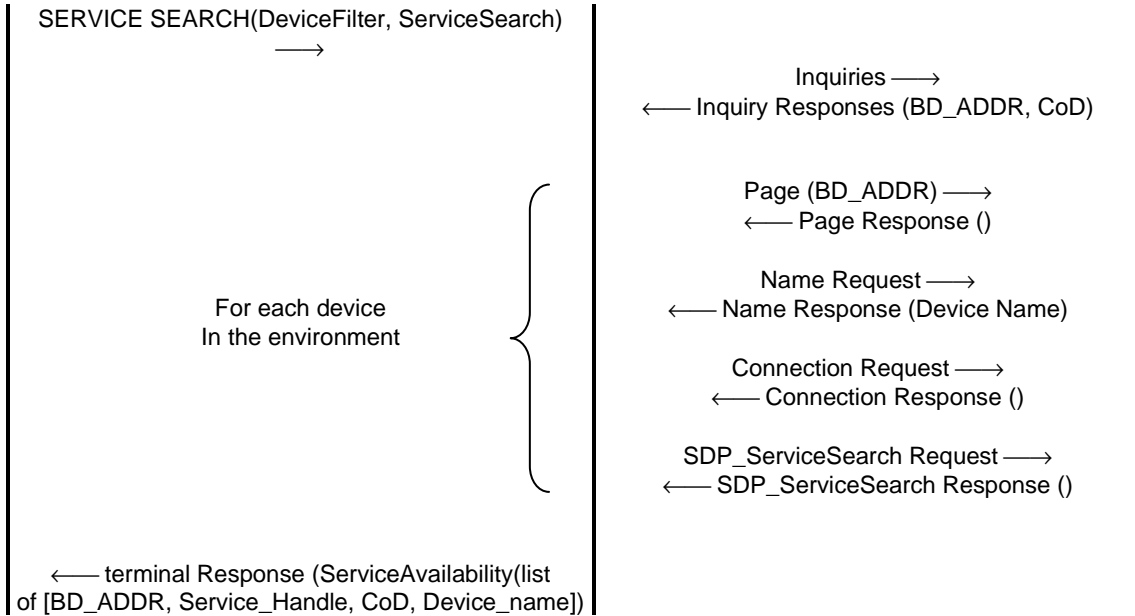
If the UICC/terminal interface parameter is not present, the UICC/terminal interface is the bearer level which is the RFCOMM level.

The Remote Entity Address shall be present and shall be the BD\_ADDR of the remote device.

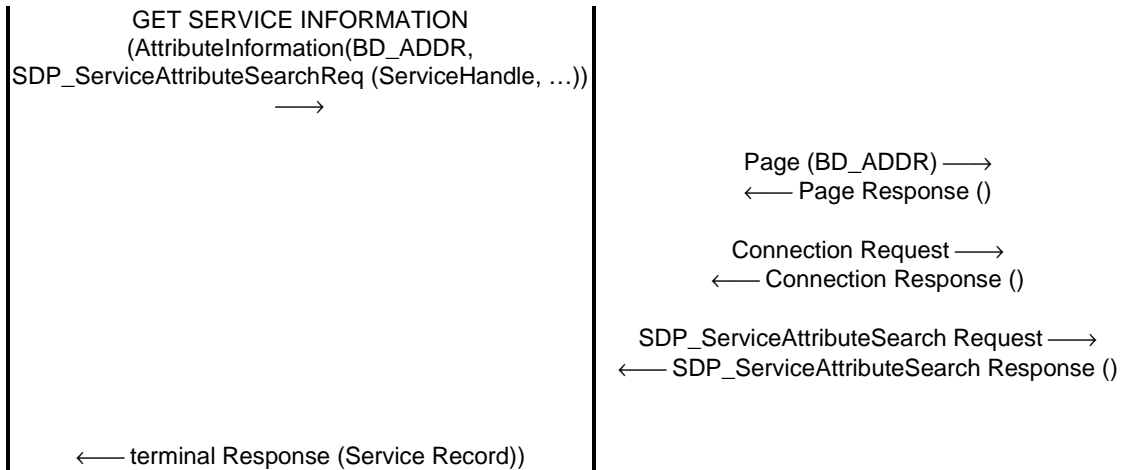
EXAMPLE: Interaction-CAT client case:

UICC	Terminal	Remote entity
------	----------	---------------

**SERVICE RETRIEVAL**



**DETAILED INFORMATION ON SERVICE**



**OPEN CHANNEL "active link establishment"**



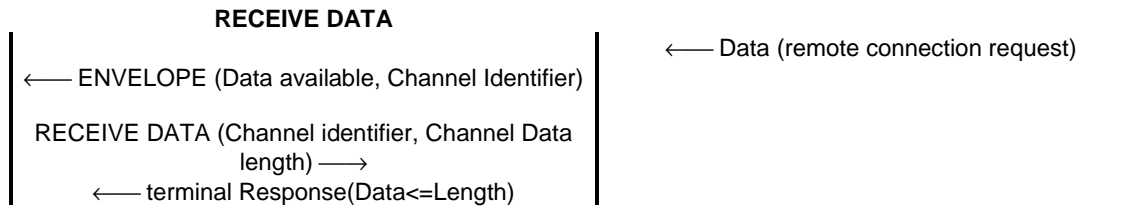
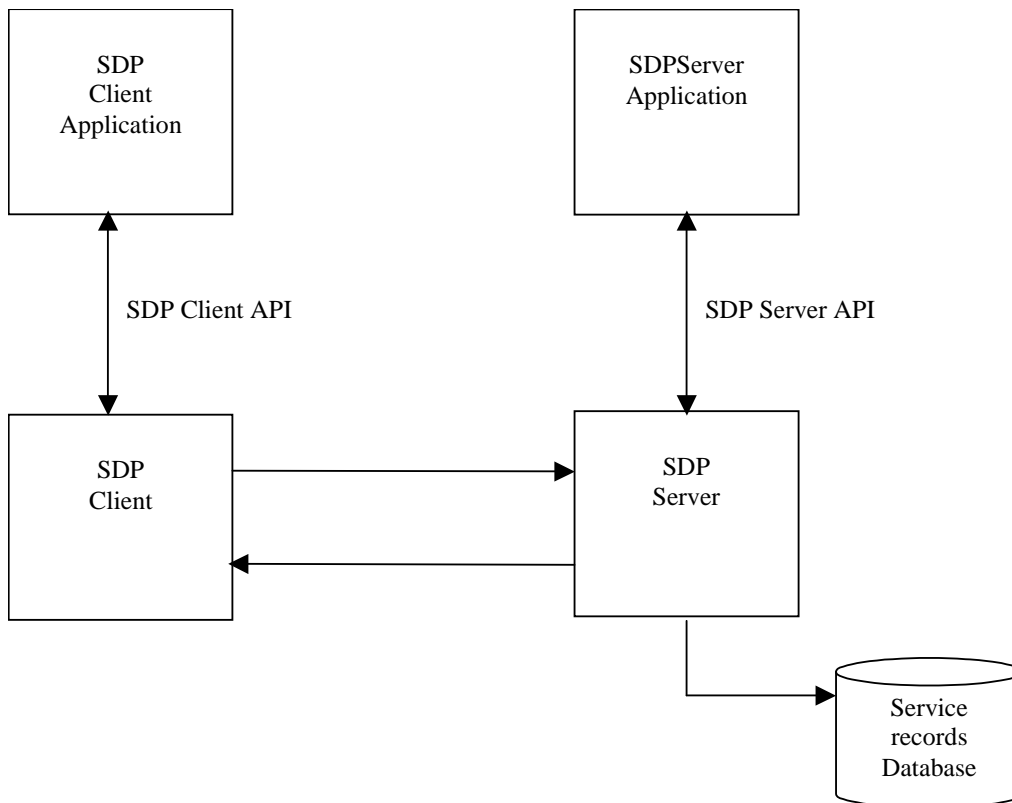


Figure K.1

## Annex L (informative): Bluetooth service discovery protocol

The service Bluetooth protocol is used to provide a way to get information of services offered by device present in a same Bluetooth environment. Each device providing a service must have a SDP Server software that can be connected by any other device. This connection is set-up by a SDP Client software and is performed in a one to one process.



**Figure L.1**

The server maintains a Service Record Database that describes the characteristics of services associated with the server. Each service record contains information about a single service. A client may retrieve information from a service record maintained by the SDP Server by issuing an SDP request.

The notion of Service Record need to be presented here for a better understanding of function set introduced. We have seen that the SDP server must maintain a list of record describing services present on the device.

The service record consists entirely of a list of service attributes.

A service record handle is a 32-bit number that uniquely identifies each service record within an SDP server.

---

## L.1 Service attribute

Each service attribute describes a single characteristic of a service. Each service attribute consists of two components: an attribute ID and an attribute value. The set of attributes characterizing one service are gathered in a service record. Table L.1 introduces examples of attributes that can be used in a service record.

**Table L.1**

ServiceClassIdList	Identifies the type of service represented by a service record. In other words, the list of classes of which the service is an instance
ServiceID	Uniquely identifies a specific instance of a service
ProtocolDescriptorList	Specifies the protocol stack(s) that may be used to utilize a service
ProviderName	The textual name of the individual or organization that provides a service
ServiceName	A text string containing a human readable name for the service
ServiceDescription	A text string describing the service

The USAT application shall provide such record to the SDP server in order to become reachable by any other device. Information shall be presented to the SDP server in the good format (see Bluetooth specification [16]) to be easily integrated in its own Service record Database.

Following is a brief description of the way by which a USAT application could retrieve a service residing on another device.

A Bluetooth device can perform a search by Patterns (Service UUID or Attributes) or by browsing. A service browsing must interact with the user. We here prefer that the USAT application simply sends a search that the SDP Client terminal software will perform. The USAT application will perform a Service Search with a service search pattern. A service search pattern is a list of UUIDs used to locate matching service records. The USAT application will prepare PDU(s) that the SDP client software will just have to push to L2CAP layer and to SDP Server software residing on another device. Once the USAT gets the list of services available, it can get further information on the services and then select one to perform an OPEN CHANNEL.

# Annex M (informative): Use of CAT bearer independent protocol for local links, server case

This annex applies to classes "e" and "f".

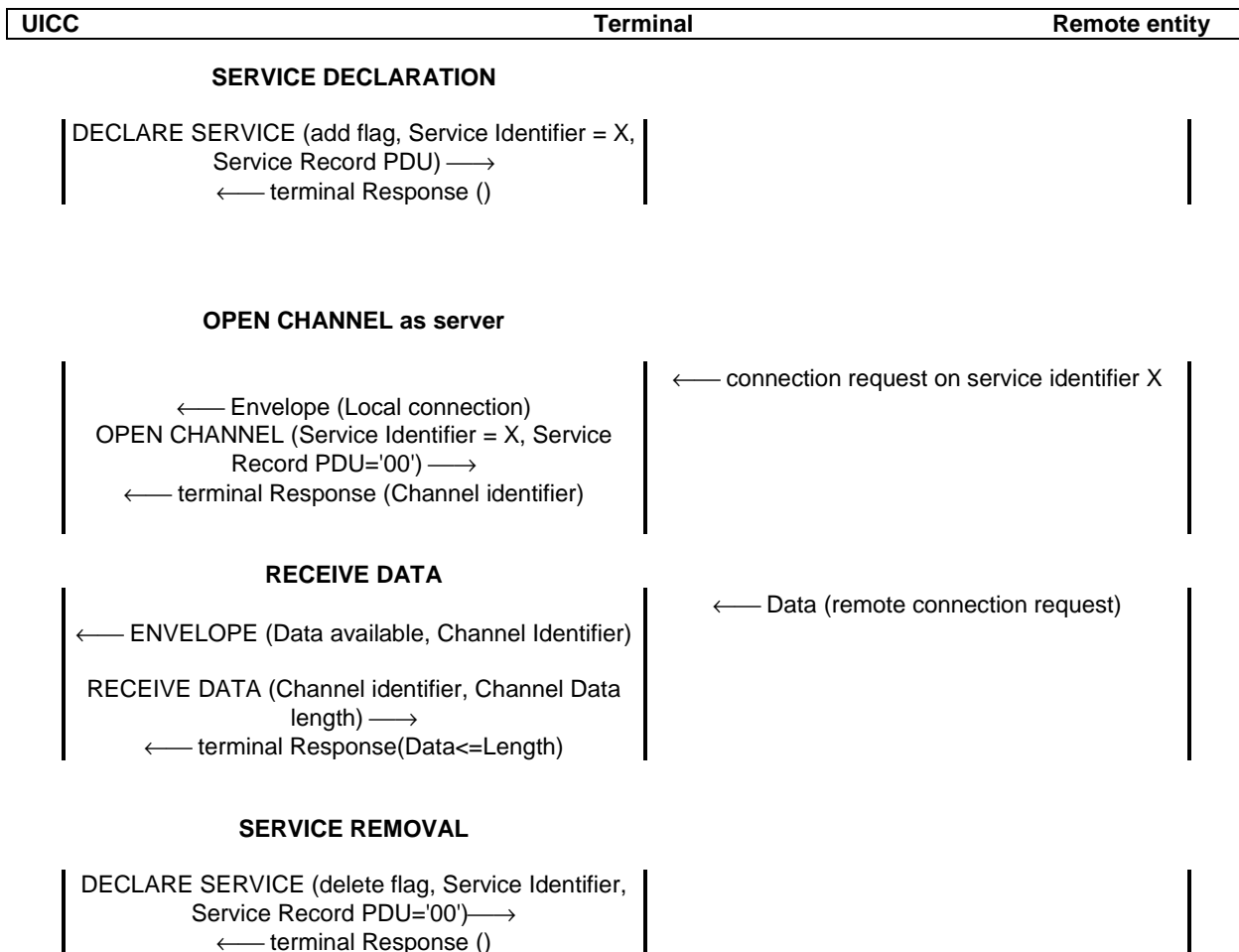


Figure M.1



---

## Annex N (informative): Browsing terminology

### References:

- WAP specifications: <http://www.wapforum.org/>.
- HTML and XHTML specifications: <http://www.w3.org/TR/>
- CHTML specifications: <http://www.w3.org/TR/1998/NOTE-compactHTML-19980209/>

### Definitions:

**WML:** Browsing language used by WAP

**HTML:** HyperText Markup Language, from the World Wide Web Consortium

**XHTML:** Extensible HyperText Markup Language, from the World Wide Web Consortium

**CHTML:** Compact HyperText Markup Language, used by I-Mode

---

## Annex O (informative): Use of USAT Bearer independent protocol for local links IrDA case

IrDA services to be run by the UICC should be developed so that the access to their service record is open and does not necessitate any security mechanism (no authentication or encryption).

---

### O.1 Service Search command

The Local Bearer Technology Identifier is IrDA. Service Search consists for the terminal in first performing a device discovery of the devices that conform to the Device Filter; then performing a LM\_GetValueByClass with the class name given in the Service Search TLV. The terminal shall then return the Service Availability data object which is a list of address + Character Set + ServiceHints + Device\_NickName.

Note for Handset Manufacturers:

As the mobile is not always connected to other devices present in the remote environment (e.g. IrDA), when performing a service search, it is up to the terminal to set a procedure that allows:

- A "scan" of the environment to discover new devices.
- A connection to Service Discovery Servers of discovered devices.
- A match with the requested service to set up the response to the USAT application.

---

### O.2 Get Service Information command

The Local Bearer Technology Identifier is IrDA, GET SERVICE INFORMATION consists for the terminal in connecting to a specific device and performing a series of LM\_GetValueByClass operations.

The terminal shall then return the Service Record data object.

When performing GET SERVICE INFORMATION, it is up to the terminal to set up a connection with the requested device and perform the IAS exchange.

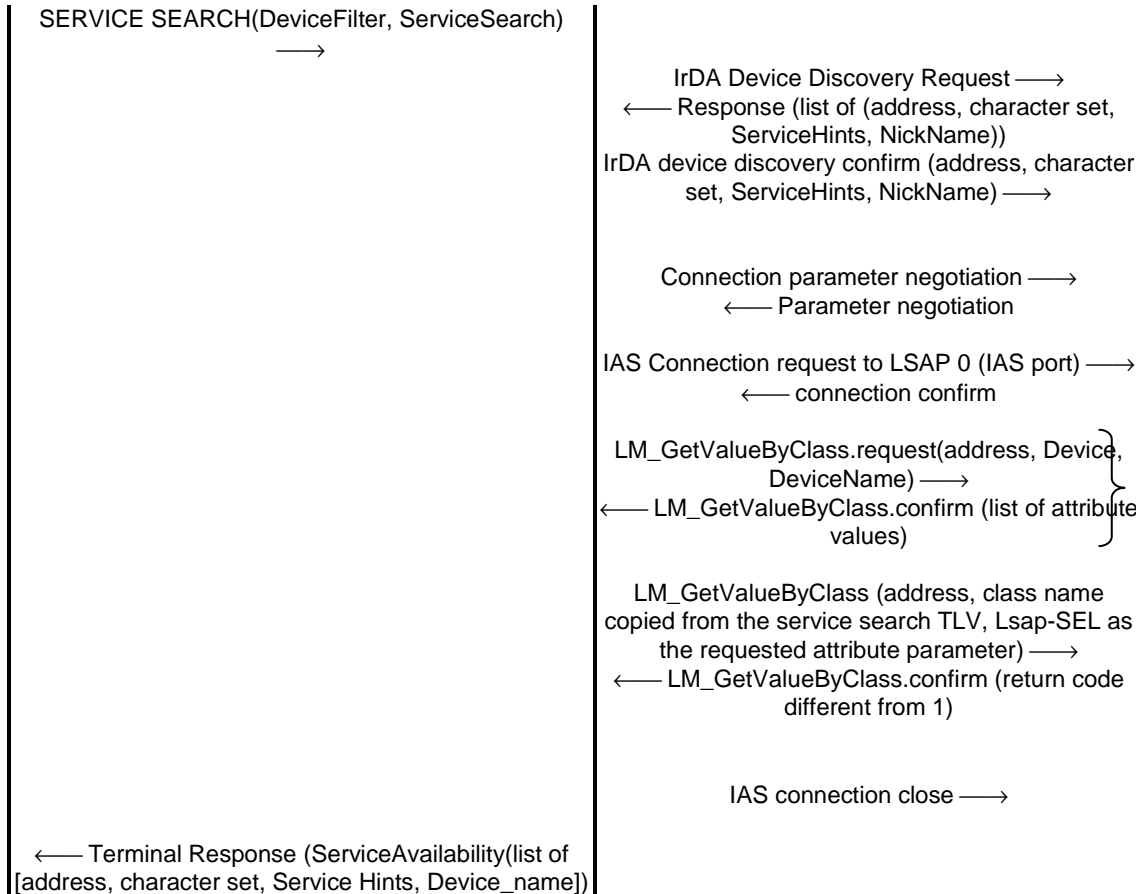
## O.3 OPEN CHANNEL command

The Remote Entity Address shall be present and shall be the 32 bits address of the remote device.

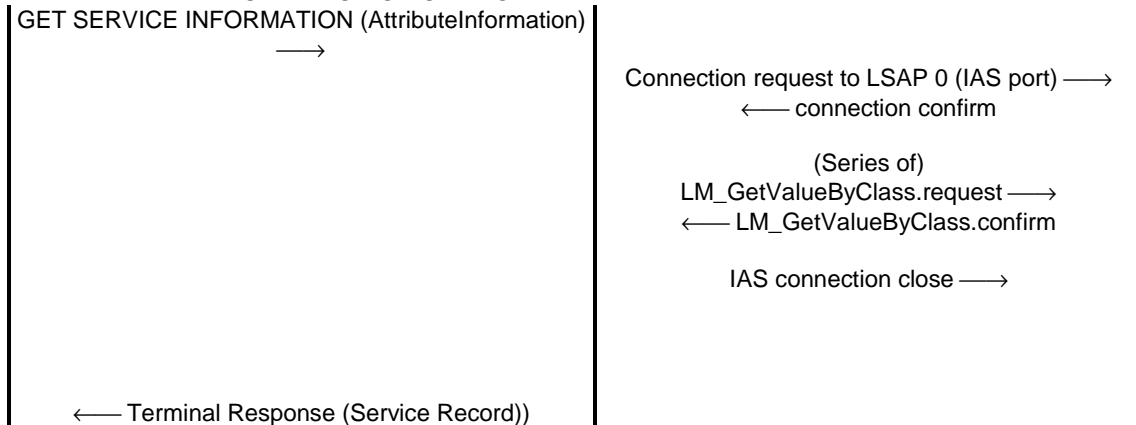
EXAMPLE: Interaction - USAT client case:

UICC	Terminal	Remote entity
------	----------	---------------

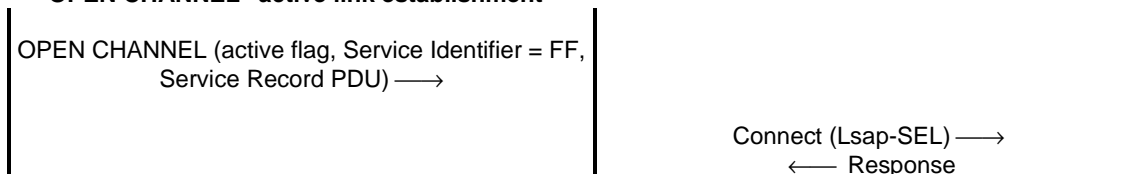
### SERVICE RETRIEVAL



### DETAILED INFORMATION ON SERVICE



### OPEN CHANNEL "active link establishment"



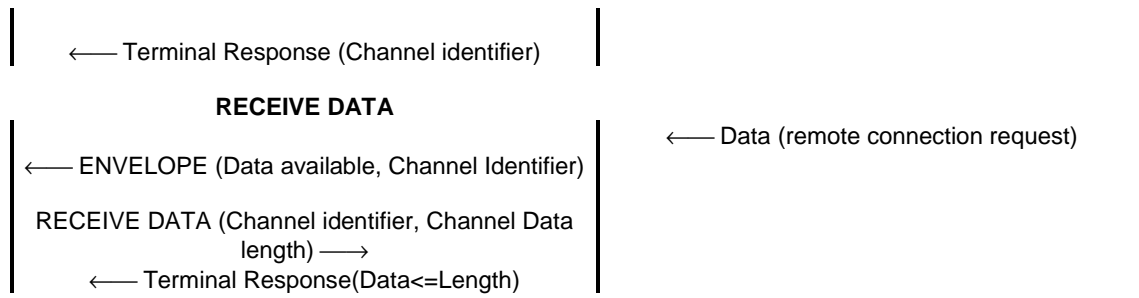


Figure O.1

# Annex P (informative): IrDA Information Access Service

The Information Access Service (IAS) maintains information about services provided by the device and also provides operations (e.g. LM\_GetValueByClass) for remotely accessing the information base on another device.

The information stored in the server information base consists of a number of objects. Each object defines a single service. It has a class name, an identifier that uniquely specifies the object within the device, and a number of attributes.

An attribute is a name-value pair. The name is a length-encoded sequence of octets. The value is a typed field, with a length field if the type is not of fixed length, and a sequence of octets comprising the actual value. Each service attribute describes a single characteristic of a service. The one essential attribute for every entry is the LSAP-SEL (or service address), which is required in order to make a LMP connection to the service. Figure P.1 shows an example of the information access service database for a device offering two unique services.

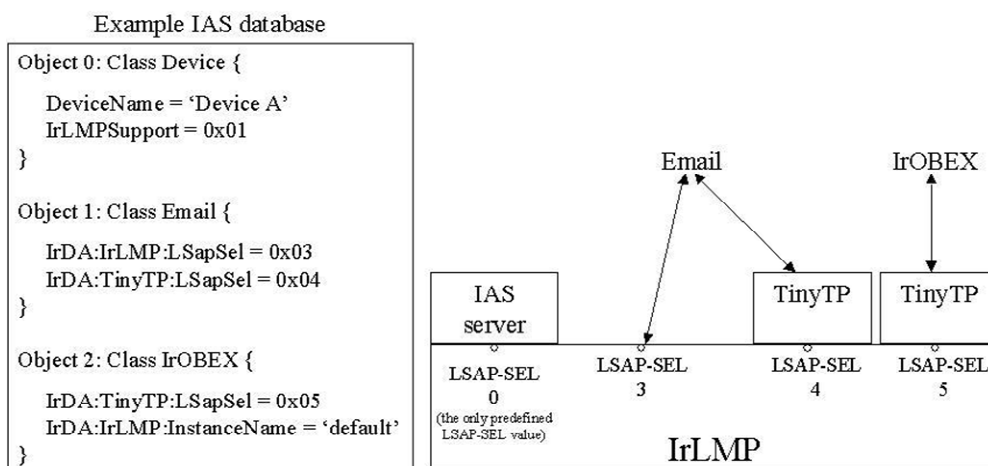


Figure P.1

The example shows a device with two individual applications: e-mail and IrOBEX (file transfer application). The information base contains two objects associated with these applications. The required Object 0 is always present within the information access service database, and it provides information about the device name, the version of IrLMP the device supports, the IAS and IrLMP primitives supported. All other devices can address Object 0 to get this information. Objects in the information base typically detail information about the services provided, for example, the LSAP-SEL where these services can be accessed. In the case of the email application, this service can be accessed using the Tiny TP flow-control mechanism on LSAP-SEL 4, or directly on LSAP-SEL 3. The difference is encoded in the attribute name.

IAS provides several service primitives to access information access service data. However, the only mandatory service is GetValueByClass. This service requires the service user to provide the class and attribute names of the service it is interested in.

The USAT application (only server applications) shall provide such record (object) to the IAS server in order to become reachable by any other device. Information shall be presented to the IAS server in the good format (see IrLMP specification [33]) to be easily integrated in its own Service record Database.

## Annex Q (informative): IrDA IAS class name and associated parameters

The client can get the remote device IAS capabilities by querying the mandatory IAS objects 0 (instance of class Device). Thus, the client can find out which IAS IrLMP primitives are supported. If all IAS primitives are supported, the client can proceed by querying the IAS server in order to discover which services are available (thus getting classes names) and to determine the characteristics of those available services (attributes names and values). The description of those primitives is given in IrLMP specification [33] in the Information Access Service clause.

If the IAS server supports only the mandatory primitive LM\_GetValueByClass, the client shall have a prior knowledge of the class name or the service name. Since the LSAP-SEL (or service address) is a mandatory attribute in all classes, the client can use it to call the LM\_GetValueByClass primitive and thus find out the presence of a service. If the return code is 1 (which means no such class) the client can proceed the same way to search for other services or wait for new devices to come in range, else conclude that the service exists.

The name of the LSAP-SEL attribute is IrDA:XX:LsapSel where XX is the transport layer. For instance, if the transport layer is TinyTP the LSAP-SEL attribute's name is IrDA:TinyTP:LsapSel. The following table presents the attributes names of the IrCOMM service (class name = "IrDA:IrCOMM") as defined by IrDA.

**Class name:** IrDA:IrCOMM

Attribute Name	Value Type	Description
IrDA:TinyTP:LsapSel	Integer (0x01)	The IrLMP LSAP/TTPSAP of the TTP entity that provides access to the service being advertised Legal values are restricted to the range 0x01-0x6F. This IAS entry is for one or more of the cooked service.
IrDA:IrLMP:LsapSel	Integer (0x01)	The IrLMP LSAP of the service being advertised Legal values are restricted to the range 0x01-0x6F. This IAS entry is for 3-Wire raw services.
Parameters	Octet seq (0x02)	A collection of one or more parameters characterizing an IrCOMM service.
IrDA:IrLMP:InstanceName	UserString (0x03)	A displayable string to help distinguish among otherwise identical IAS objects.

The parameters attributes contain one or more values (themselves called parameters), which characterize the service being provided. Each parameter in the Parameters attribute consists of a 3-tuple (tag, length and value).

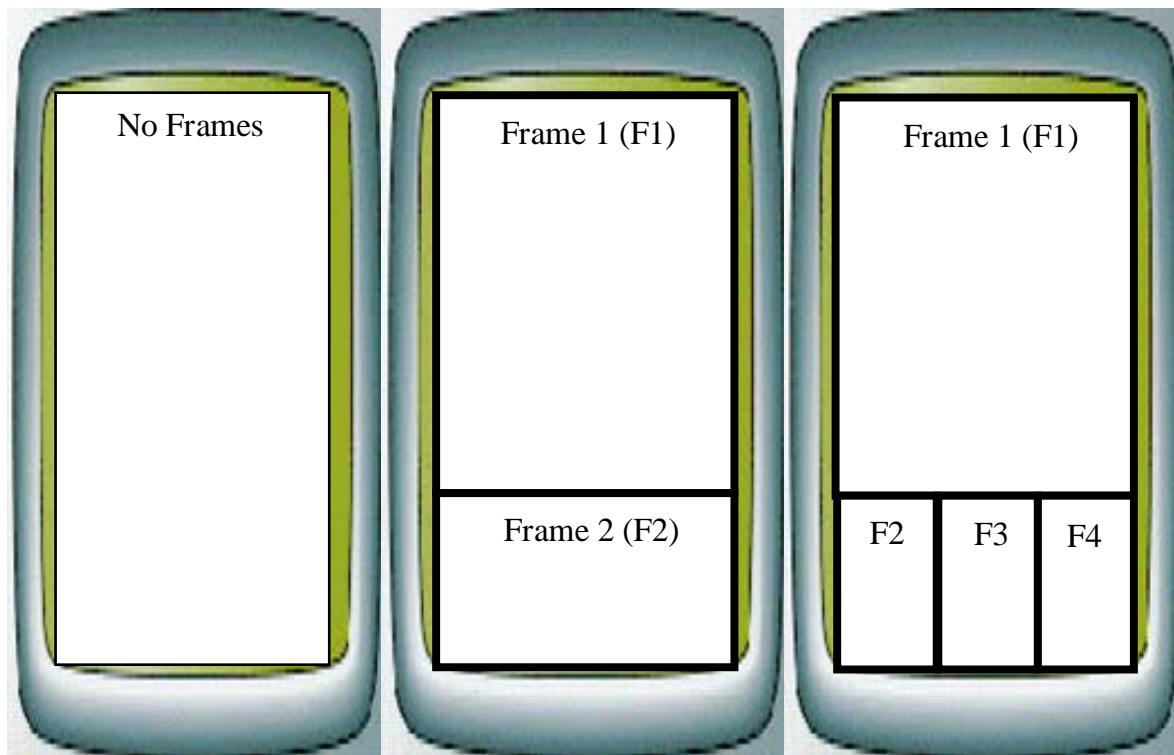
The Parameters attribute collects into one place many characteristics, which together define a service. The same information could have been spread into multiple attributes, but that would require multiple IAS GetValueByClass queries, an implementation inconvenience. On other hand the client does not have to discover attributes names.

## Annex R (informative): Example of Frames usage

The following example shows the use of the commands of the frames in conjunction with the LAUNCH BROWSER and DISPLAY TEXT.

Frames setup

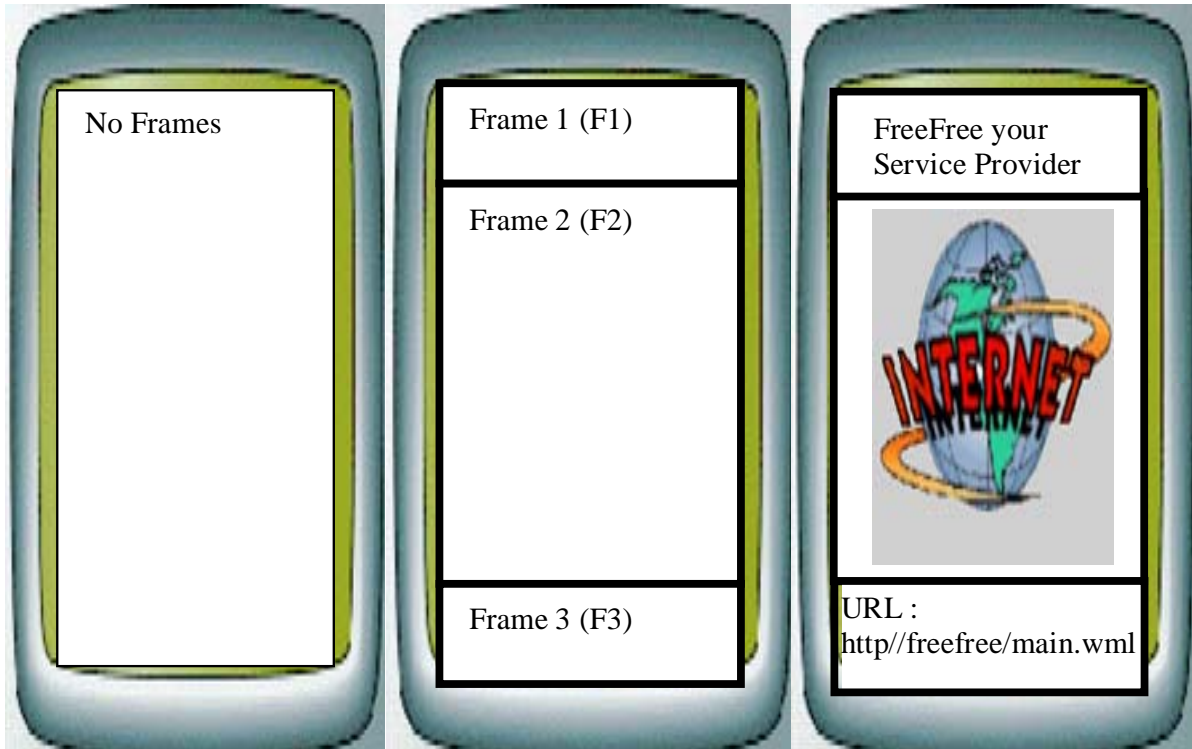
<b>terminal</b>		<b>UICC</b>
TERMINAL PROFILE	----->	
	<-----	91 XX
FETCH	----->	
	<-----	SET FRAMES (0,horizontal,2,1)
TR (OK, Frame Info ID=1, Frame Info ID=2)	----->	
	<-----	91 XX
FETCH	----->	
	<-----	SET FRAMES (2,vertical,1,1,1)
TR (OK, Frame Info ID=1, Frame Info ID=2, Frame Info ID=3, Frame Info ID=4)	----->	
	<-----	91 XX
FETCH	----->	
	<-----	GET FRAMES STATUS ()
TR (OK, Frame Info ID=1, Frame Info ID=2, Frame Info ID=3, Frame Info ID=4)	----->	
	<-----	90 00



Use case

terminal		UICC
TERMINAL PROFILE	----->	
	<-----	91 XX
FETCH	----->	
	<-----	SET FRAMES (0,horizontal,1,2,1)
TR (OK, Frame Info ID=1, Frame Info ID=2, Frame Info ID=3)	----->	
	<-----	91 XX
FETCH	----->	
	<-----	DISPLAY TEXT sustained(Frame 1, "FreeFree Your Service Provider")
TERMINAL RESPONSE (OK)	----->	
	<-----	91 XX
FETCH	----->	
	<-----	DISPLAY TEXT sustained(Frame 3, "URL: <a href="http://freefree/main.wml">http://freefree/main.wml</a> ")
TERMINAL RESPONSE (OK)	----->	
	<-----	91 XX
FETCH	----->	
	<-----	LAUNCH BROWSER (Frame 2, URL)
TERMINAL RESPONSE (OK)	----->	
	<-----	91 XX





## Annex S (normative): Support of CAT by Terminals with reduced feature capabilities.

Terminals may be designed for specialized uses and may therefore not support all functions due to reduced feature capabilities.

If a terminal states conformance to a terminal type defined in this annex, some CAT functionalities will have restrictions of usage, or will not be able to be used for this type of terminals, due to the specificities (i.e. reduced feature capabilities) of the terminal.

Table S.1 defines different types of Terminals with reduced features capabilities.

**Table S.1: Definition of terminal types**

Terminal type	Type definition
ND	Terminal that has no display capability (see note 1)
NK	Terminal that has no keypad available (see note 2)
NA	Terminal that has no audio alerting capability
NS	Terminal that has no speech call capability
NL	Terminal that does not support multiple languages
NOTE 1: No display capability means no ability to display something to the user.	
NOTE 2: No keypad available means no capability to interact (i.e. input or confirmation) with the user via a MMI.	

A terminal of type ND shall set "no display capability" in the terminal profile.

A terminal of type NK shall set "no keypad available" in the terminal profile.

A terminal may indicate several terminal types (i.e. several reduced feature capabilities).

If a keyboard or display or alerter is connected to the device after the Terminal Profile has been issued, the Terminal may resend the Terminal Profile indicating this change.

Table S.2 provides the applicability of envelope commands for the different terminal types.

Table S.3 provides an overview of affected commands.

Table S.2: Envelope applicability table

Envelope	ND type	NK type	NA type	NS type	NL type
MENU SELECTION	O	O			
CALL CONTROL	Note 2			O	
TIMER EXPIRATION					
EVENT DOWNLOAD - MT call				O	
EVENT DOWNLOAD - Call connected				O	
EVENT DOWNLOAD - Call disconnected				O	
EVENT DOWNLOAD - Location status					
EVENT DOWNLOAD - User activity		O			
EVENT DOWNLOAD - Idle screen available	O				
EVENT DOWNLOAD - Card reader status					
EVENT DOWNLOAD - Language selection					O
EVENT DOWNLOAD - Browser termination	O	O			
EVENT DOWNLOAD - Data available					
EVENT DOWNLOAD - Channel status					
EVENT DOWNLOAD - Access Technology Change					
EVENT DOWNLOAD - Display parameters changed	O				
EVENT DOWNLOAD - Local Connection					
EVENT DOWNLOAD - Network Search Mode Change					
EVENT DOWNLOAD - Browsing status	O	O			
EVENT DOWNLOAD - Frames Information changed	O				
MMS Transfer Status					
MMS notification download					
TERMINAL APPLICATIONS					
NOTE 1: "O" means proactive command is optional, No indication means that the proactive command is fully applicable.					
NOTE 2: If an alpha identifier is provided by the UICC in the response, it shall be ignored by the terminal.					

Table S.3: Overview of affected commands

Command	ND type	NK type	NA type	NS type	NL type
DISPLAY TEXT	O	partial			
GET INKEY	O	O			
GET INPUT	O	O			
MORE TIME					
PLAY TONE	partial		O		
POLL INTERVALL					
REFRESH	partial				
SET UP MENU	O	O			
SELECT ITEM	O	O			
SEND SHORT MESSAGE	partial				
SET UP CALL	O	O		O	
POLLING OFF					
PROVIDE LOCAL INFORMATION					
SET UP EVENT LIST					
PERFORM CARD APDU					
POWER OFF CARD					
POWER ON CARD					
GET READER STATUS					
TIMER MANAGEMENT					
SET UP IDLE MODE TEXT					
RUN AT COMMAND	partial				
SEND DTMF	partial			O	
LANGUAGE NOTIFICATION					O
LAUNCH BROWSER	O	O			
OPEN CHANNEL related to CS bearer	partial	partial			
OPEN CHANNEL related to packet data service bearer	partial	partial			
OPEN CHANNEL related to local bearer	partial	partial			
OPEN CHANNEL related to Default (network) bearer	partial	partial			
OPEN CHANNEL related to UICC Server Mode	partial				
OPEN CHANNEL related to Terminal Server Mode					
CLOSE CHANNEL	partial				
RECEIVE DATA	partial				
SEND DATA	partial				
GET CHANNEL STATUS					
SERVICE SEARCH	partial				
GET SERVICE INFORMATION	partial				
DECLARE SERVICE					
SET FRAMES	O				
GET FRAME STATUS	O				
RETRIEVE MULTIMEDIA MESSAGE	partial				
SUBMIT MULTIMEDIA MESSAGE	partial				
DISPLAY MULTIMEDIA MESSAGE	O	partial			
ACTIVATE					

NOTE: "O" means support of this command is optional, "partial" means parts of the command are affected. No indication means that the proactive command is fully applicable.

## Annex T (normative): Default routing for CAT over the modem interface

The present Annex applies only if the terminal supports class "s".

In case a modem supports routing of CAT to a Connected Entity (e.g. via AT commands as specified in TS 127 007 [5]) or in case several CAT clients within the Connected Entity provide CAT facilities, care has to be taken that the facilities provided by each entity for default routing are unique. In case two entities provide support for the same facility, a conflict is detected.

The only facilities indicated in the TERMINAL PROFILE which may be supported by multiple entities at the same time without conflict are given in table T.1.

**Table T.1: Facilities that may be supported by multiple entities**

Facility	Remarks
Profile download	
Command result	
No display capability (i.e. class "ND" is indicated)	
No keypad available (i.e. class "NK" is indicated)	
Proactive UICC: REFRESH	
Proactive UICC: SET UP EVENT LIST	
Event: Data available	Note 2
Event: Channel status	Note 2
Event: Local connection	Note 2
Proactive UICC: OPEN CHANNEL	Note 1
Proactive UICC: CLOSE CHANNEL	Note 2
Proactive UICC: RECEIVE DATA	Note 2
Proactive UICC: SEND DATA	Note 2
Proactive UICC: GET CHANNEL STATUS	Note 2
Proactive UICC: SERVICE SEARCH	Note 2
Proactive UICC: GET SERVICE INFORMATION	Note 2
Proactive UICC: DECLARE SERVICE	Note 2
Number of channels supported by terminal	Note 3
TCP, UICC in client mode, remote connection	Note 2
UDP, UICC in client mode, remote connection	Note 2
Profile Container, Envelope Container, COMMAND CONTAINER and ENCAPSULATED SESSION CONTROL	Note 4
NOTE 1: Uniqueness is provided by means of the bearer type.	
NOTE 2: Uniqueness is provided by means of the channel identifier.	
NOTE 3: The total number of channels supported shall be sum of the respective number of supported channels by each entity, limited to a maximum of 7.	
NOTE 4: Uniqueness of Profile Container is provided by means of the eCAT client name. Uniqueness of the other eCAT commands and events is provided by means of the eCAT client identifier.	

Some specific facilities indicated in the TERMINAL PROFILE can be supported only by the modem. In case a Connected Entity provides support for any of those facilities, a conflict is detected. Unless an NAA defines its own list, the list of such facilities is as follows:

- Call control by NAA.
- MO short message control support.
- PROVIDE LOCAL INFORMATION (MCC, MNC, LAC, Cell ID & IMEI).
- PROVIDE LOCAL INFORMATION (NMR).
- POLL INTERVAL.
- POLLING OFF.

- Event: MT call.
- Event: Call connected.
- Event: Call disconnected.
- Event: Location status.
- Event: Access Technology Change.
- Event: Network Search Mode Change.
- PROVIDE LOCAL INFORMATION (ESN).
- Call control on GPRS.
- PROVIDE LOCAL INFORMATION (IMEISV).
- PROVIDE LOCAL INFORMATION (Search Mode change).
- PROVIDE LOCAL INFORMATION (NMR(UTRAN)).
- Steering of Roaming for I-WLAN REFRESH support.

---

## T.1 Default routing mechanism

The modem shall forward proactive commands that it does not process itself to the Connected Entity. In a similar way, these commands shall be distributed within the Connected Entity if it consists of multiple CAT clients.

For CAT facilities that are supported by more than one entity the following shall apply:

- SET UP EVENT LIST shall be routed to all entities supporting the command, each containing only the events supported by the entity, even if the list is empty (which allows for proper deregistration of events set up earlier). For the TERMINAL RESPONSE to the UICC, the responses from the modem and the Connected Entity have to be combined as follows:
  - The modem shall check if it is able to set up the events it supports itself. If the modem is currently unable to process command or if the set up of the events would fail, the modem shall send this result in the TERMINAL RESPONSE without forwarding the command to the Connected Entity.
  - If the modem is capable of setting up the modem events, the list of Connected Entity events shall be forwarded to the Connected Entity and the Connected Entity shall send its TERMINAL RESPONSE.
  - If the Connected Entity command was successful, the modem shall set up its events and report that the command was performed in the TERMINAL RESPONSE. If the modem or the Connected Entity or both have performed the command with partial comprehension or with missing information, this shall be reflected in the TERMINAL RESPONSE; if one reported partial comprehension and the other missing information, the modem response takes precedence.
  - If the Connected Entity reports that it is currently unable to process command or the command failed, the modem shall report this in the TERMINAL RESPONSE.
- REFRESH shall be routed to all entities supporting the command to inform them about modified EFs; only the modem shall perform other activities indicated in the command (e.g. UICC reset). For the TERMINAL RESPONSE to the UICC, the responses from the modem and the Connected Entity have to be combined as follows:
  - The modem shall check if it is able to perform the REFRESH. If the modem is currently unable to process the command or the command would fail, the modem shall send this result in the TERMINAL RESPONSE without forwarding the command to the Connected Entity.

- If the modem is capable of performing the REFRESH, the command shall be forwarded to the Connected Entity and the Connected Entity shall send its TERMINAL RESPONSE, but if there is a refresh action to be performed by the modem (e.g. NAA initialization), the modem shall send its response to the Connected Entity's TERMINAL RESPONSE only after the refresh action has started to avoid that the Connected Entity tries to access the UICC before the refresh action.
  - If the Connected Entity command was successful, the modem shall perform the REFRESH and report that the command was performed in the TERMINAL RESPONSE. If the modem or the Connected Entity have performed the command with a limitation (partial comprehension, missing information, additional EFs read, requested icon could not be displayed or NAA was not active) this shall be reflected in the TERMINAL RESPONSE; if both reported different limitations, the modem's response takes precedence.
  - If the Connected Entity reports that it is currently unable to process command or the command failed, the modem shall report this in the TERMINAL RESPONSE.
- OPEN CHANNEL shall be routed according to the indicated bearer type. To avoid conflicts in channel identifier assignment, the modem shall replace the destination device identity by an available channel identifier and the Connected Entity shall use this channel identifier in its response.
  - Subsequent BIP commands shall be routed according to the channel identifier.
  - eCAT commands shall be routed according to the eCAT client identifier.

---

## T.2 Combination rules for terminal profiles

The modem shall combine the facilities provided by the modem and the Connected Entity by ORing the facilities provided by both entities, except for the following:

- Number of channels supported by terminal for BIP: Here the indicated numbers of the different entities shall be added and the sum, limited to a maximum of 7, shall be provided in the combined terminal profile.

If the Connected Entity contains several CAT clients, it shall combine the facilities provided by all its CAT clients with the rule described above before sending its Terminal Profile to the modem.

## Annex U (informative): Change history

The table below indicates changes that have been incorporated into the present document since it was created by TC SCP.

Change history								
Date	Meeting	TC SCP Doc.	CR	Rv	Cat	Subject/Comment	Old	New
2000-11	SCP-03	9-00-0428	-			Initial draft based on 3GPP TS 31.111 V4.0.0 with modifications introduced by rapporteur		0.0.0
2000-12	SCP-04	SCP-010031	-			Outcome of TC SCP ad hoc meeting on CAT in December 2000. Submitted to TC SCP #4 for information and discussion	0.0.0	0.1.0
2001-02	-	SCP-010101	-			Outcome of TC SCP ad hoc meeting on CAT in February 2001. Alignment with 3GPP TS 31.111 V4.1.0	0.1.0	0.2.0
2001-03	SCP-05	SCP-010105	-			Result of drafting session during TC SCP #5	0.2.0	0.3.0
2001-03	SCP-05	SCP-010147	-			Editorial and formatting enhancements. Version number raised to 1.0.0 in line with decision at TC SCP #5	0.2.0	1.0.0
2001-06	SCP-06	SCP-010203	-			Outcome of TC SCP #6. Alignment with 3GPP TS 31.111 V4.3.0	1.0.0	1.1.0
2001-07	-	-	-			First editorial clean up following approval at TC SCP #6. This version has been sent to the ETSI secretariat for publication in July 2001 as TS 102 223 V4.0.0. No technical changes compared to V1.1.0	1.1.0	1.1.1
2001-07	-	-	-			ETSI Publication - no technical changes compared to V1.1.1	1.1.1	4.0.0
2001-10	SCP-07	SCP-010260	001		B	Terminal Profile Harmonization with ANSI/TIA/EIA-136	4.0.0	4.1.0
		SCP-010279	002		F	Corrections to the OPEN CHANNEL command		
		SCP-010280	003		F	Corrections to SEND DATA commands and Channel Status Event		
		SCP-010304	004		F	Reservations for TIA/IS-820-A facilities		
2002-01	SCP-08	SCP-010386	005		F	Corrections related to the Bearer Independent Protocol	4.1.0	4.2.0
		SCP-010384	006		F	Addition of a generic SEND SM command		
2002-06	SCP-10	SCP-020171	007	1	F	Correction of Terminal Response references	4.2.0	5.0.0
		SCP-020110	008		F	Correction of byte 17 in Terminal Profile		
		SCP-020111	009		F	Usage of Simple TLV Tag Values		
		SCP-020112	010		F	Reservation of tags&values for 3GPP2		
		SCP-020115	013		F	Correction to OPEN CHANNEL for packet data service		
		SCP-020172	011	1	B	CAT Display Menus in Colour and Various Text Formats		
2003-01	SCP-12	SCP-030070	019	1	A	Set of corrections	5.0.0	6.0.0
		SCP-030072	018	1	A	Corrections and editorial modifications		
		SCP-030032	015		A	Clarification of the usage of busy status response for Event download and Menu Selection Envelope.		
		SCP-030080	021		D	Reference TS 101 220 for definition of TLV		
2003-05	SCP-13	SCP-030117	024		A	Collection of alignments with 3GPP	6.0.0	6.1.0
2003-09	SCP-14	SCP-030287	034		A	Correction of ENVELOPE(Call Control) structure description	6.1.0	6.2.0
		SCP-030211	027		A	Correction of Item Icon Identifier list coding example		
2003-12	SCP-15	SCP-030485	043	1	B	Request of IMEISV in PROVIDE LOCAL INFORMATION	6.2.0	6.3.0
		SCP-030476	044	1	C	Extension of the "Launch Browser" capability		
		SCP-030477	050	1	B	Open Channel in Background Mode		
		SCP3-030261	042		D	Wrong reference to channel data in the TERMINAL RESPONSE table		
		SCP3-030289	029	1	B	Enable more than one text attribute per text string		
		SCP3-030291	052		A	Clarification of multiple card operation		
2004-02	SCP-16	SCP3-040052	049	1	A	Clarification on Default Bearer Description	6.3.0	6.4.0
		SCP3-040082	061		A	Clarification on user confirmation for OPEN CHANNEL		
		SCP3-040062	066		A	Correction of DECLARE SERVICE command		
		SCP3-040074	071		A	Inconsistencies in the description of the Call Connected event and in the length of the UICC/terminal interface transport level		
		SCP3-040067	033	1	B	IrDA use with BIP		
		SCP3-040066	045	2	C	Modify the TERMINAL PROFILE procedure.		
		SCP3-040065	055	2	B	Network Information envelope		
		SCP3-040061	067	1	B	Query Battery State		
		SCP3-040075	068		B	Additional information for launch browser problem		
		SCP3-040087	072		C	Removal of EIA/TIA-136 Tags		



Change history								
Date	Meeting	TC SCP Doc.	CR	Rv	Cat	Subject/Comment	Old	New
2004-05	SCP-17	SCP-040213	084		F	Correction of reference for IMEI and IMEISV	6.4.0	6.5.0
		SCP-040213	073		D	Correction of wording for call control		
		SCP-040213	076		A	Alignment of Alpha Identifier for BIP Commands		
		SCP-040213	077		F	Corrections and alignment for the Terminal Profile		
		SCP-040213	079		F	Addition of ability to set up Multi-media calls using the SET UP CALL command		
		SCP-040274	080	1		Extension of the PLAY TONE command		
		SCP-040213	081		D	Indication error between UICC and SIM		
		SCP-040213	083		D	Transfer of the COMPREHENSION-TLV Tags to TS 101 220		
		SCP-040213	082	1	F	Addition in Terminal Profile		
		2004-09	SCP-18	SCP-040312	085			
	078			1	A	Support of TCP connections for BIP local bearers		
	087				D	Editorial Correction		
	092				A	Corrections to TERMINAL RESPONSE for OPEN CHANNEL commands		
	093				D	Terminal Profile clarification		
	094				F	Correction of local links parameter for Bluetooth		
	046			3	B	Introduction of the frames in CAT		
SCP-040373	096				F	Bytes reservation for events in the Terminal Profile		
SCP-040353	095				F	Bit reservation in Terminal Profile for 3GPP features		
2004-11	SCP-19			SCP-040416	099		A	Correction of the references description
			102		A	Correction of description of DISPLAY TEXT for sustained mode		
			103		F	Suppression of the BER-TLV tags in terminal to UICC direction table and addition of the reference to TS 101 220		
			104		F	Correction of OPEN CHANNEL in background mode		
			106		F	Alignments with TS 31.111		
		SCP-040477	086	3	A	Text Attribute capabilities information		
2004-11	SCP-19	SCP-040481	107		B	Add BC repeat indicator to call control command data	6.7.0	7.0.0
2005-09	SCP-22	SCP-050242	115		A	Behaviour in case of user cancellation for BIP	7.0.0	7.1.0
			117		C	Alignment of user confirmation phase for toolkit commands		
			122		A	Remove clause 11 "Security Requirements"		
			124		A	Introduction of MEID		
			119		A	SETUP CALL and outgoing call information storage		
		SCP-050284	121		A	Bit reservation for 3GPP in Terminal Profile.		
		SCP-050294	127		A	Correction of description of command SET FRAMES		
		SCP-050229	129		D	Clarification of Channel identifier for BIP		
			130		B	Transfer MMS commands from 3GPP		
			132		A	Correction of refresh in case of multiple running applications		
	SCP-050281	128	1	F	Clarification to the support of text attributes in the Terminal Profile by the ME			
2005-12	SCP-23	SCP-050446	139		B	BIP acting in server mode	7.1.0	7.2.0
		SCP-050462	135		A	Correction of the definition of the Item text attribute list comprehension TLV		
			137		A	Clarification of Refresh mode		
		SCP-050502	136		B	Reservation of values for 3GPP concerning the I-WLAN bearer		
		SCP-050504	139		D	Wording alignment for bits reservation in Terminal Profile		
2006-03	SCP-25	SCP-060130	141		A	Include browsing status event in letter class "c"	7.2.0	7.3.0 withdrawn
			142		F	Introduction of Alpha Identifier support in the REFRESH command		
			143		F	Clarification of the usage of the CLOSED status when BIP is acting in server mode		
2006-07	SCP-26	SCP-060285	145		A	Correction of incorrect references on 3GPP2 specifications; Rel-7	7.2.0	7.4.0
		SCP-060256	146	1	A	Bit reservation for 3GPP in Terminal Profile		
			147	1	F	Clarification of the UICC refresh mode in the REFRESH command		
			148	12	F	Add missing bit in Terminal Profile for "TCP, UICC in server mode"		
2006-09	SCP-27	SCP-060442	149		F	Addition of missing TLVs in the OPEN CHANNEL related to UICC Server Mode	7.4.0	7.5.0
		SCP-060469	150	1	F	Discarding of out of date HTTP data.		
2007-01	SCP-29	SCP-070031	152		D	Insertion of note on maximum length in GET INPUT	7.5.0	7.6.0

Change history								
Date	Meeting	TC SCP Doc.	CR	Rv	Cat	Subject/Comment	Old	New
2007-05	SCP-30	SCP-070136	153		D	Correction of badly formulated sentence for Call control and emergency call	7.6.0	7.7.0
			154		D	Clarification of terminal response for Refresh		
			155		C	Introduction of 'predictive text' mode into GET INPUT proactive command		
	SCP-30bis	SCP-070189	156		B	Addition of clarifications to enable support of CAT Applications over the USB Interface		
2007-07	SCP-31	SCP-070275	151		B	Set of commands for the Launch Application feature	7.7.0	7.8.0
			157		B	Card request to have the TCP connection go into "TCP in LISTEN state"		
2007-08						Correction to implementation error of CR151 whereby clause 6.6.27.6 was wrongly inserted after 6.4.27.6.	7.8.0	7.8.1
2007-08	SCP-32	SCP-070316	158	1	C	Alignment with 3GPP: Steering of Roaming Refresh Command	7.8.1	7.9.0
			160		F	Clarification the differences between proactive UICC sessions and proactive UICC application sessions		
2007-10	SCP-33	SCP-070422	161	-	B	Introduction of an interface activation proactive command following the approval of TS 102 613 (UICC-CLF interface)	7.9.0	7.10.0
2008-01	SCP-35	SCP-080017	163		F	Correction of reference to 3GPP TS 23.140	7.10.0	7.11.0
2008-04	SCP-37	SCP-080210	164		F	Correction for Terminal Server Mode	7.10.0	7.11.0
2008-01	SCP-35	SCP-080017	162	1	B	Alignment with 3GPP: New Network Rejection event	7.11.0	8.0.0
2008-07	SCP-38	SCP-080314	165		B	Reserve Type of Command values for proprietary use	8.0.0	8.1.0
		SCP-080314	168		A	Essential editorial corrections	8.0.0	8.1.0
		SCP-080314	170		A	Addition of missing Frame Identifiers in the MMS commands	8.0.0	8.1.0
2008-07	SCP-39	SCP-080440	171	1	B	Alignment with 3GPP for Geographical Location discovery mechanism	8.1.0	8.2.0
		SCP-080440	172	1	C	Code reservation for Device Management and Device Synchronization applications in Registry Application Data	8.1.0	8.2.0
		SCP-080440	173	1	B	Addition of Location Information for Broadcast technologies	8.1.0	8.2.0
		SCP-080440	174		A	Addition of HCI connectivity event	8.1.0	8.2.0
2009-01	SCP-40	SCP-090065	176	1	B	Reduced capability terminals in CAT	8.2.0	8.3.0
		SCP-090022	177		A	Addition of HCI connectivity event in Section 10	8.2.0	8.3.0
		SCP-090022	178		B	Reservation of values for 3GPP related to I-WLAN Steering of Roaming Refresh Command	8.2.0	8.3.0
2009-04	SCP-41	SCP-090116	183		A	Downgrade of USSD Data download and application mode to a letter class	8.3.0	8.4.0
		SCP-090116	179		F	Addition of Access Technologies	8.3.0	8.4.0
		SCP-090137	184		F	Terminal Profile bits allocation for 3GPP (support of LTE in 3GPP TS 31.111)	8.3.0	8.4.0
		SCP-090168	180	1	C	Addition of APN settings objects in the Launch Browser command	8.4.0	9.0.0
2009-08	SCP-43	SCP-090324	187		A	Bearer type reservation for 3GPP	9.0.0	9.1.0
		SCP-090353	185	1	B	Enabling and disabling contactless functionality of the UICC	9.0.0	9.1.0
2010-03	SCP-44	SCP(10)0083	188	1	D	Collection of corrections	9.0.0	9.1.0
		SCP(10)0084	190	-	D	Correction of terminology	9.0.0	9.1.0
-	-	-	-	-	-	Alignment of Terminal Profile structure with 3GPP TS 31.111 (CSG cell selection event)	9.0.0	9.1.0
2010-07	SCP-45	SCP(10)0149	193		A	Cleanup of errors in BIP sections	9.1.0	9.2.0
		SCP(10)0169	198	1	A	Correction of length in registry application data	9.1.0	9.2.0
		SCP(10)0170	199	1	C	Addition of objects for user confirmation	9.2.0	10.0.0
2010-10	SCP-46	SCP(10)0252r1	202	1	A	Correction of NMR for multiple access technologies	10.0.0	10.1.0
		SCP(10)0253	203		B	Direct application channel to terminal applications	10.0.0	10.1.0
		SCP(10)0257	207		A	Missing parameter values for Terminal Server mode	10.0.0	10.1.0
		SCP(10)0258	208		B	Addition of launch parameters for terminal applications	10.0.0	10.1.0
		SCP(10)0259	209		B	Extension of Broadcast Network Information for WiMAX and FLO	10.0.0	10.1.0
		SCP(10)0262	212		A	Correction of Byte(s) column in Broadcast Network Information	10.0.0	10.1.0
2011-01	SCP-47	SCP(11)0040	215		A	Correction to terminal profile bit for 3GPP	10.1.0	10.2.0
		SCP(11)0043	216		B	Reservation of terminal profile bit for 3GPP (IMS Communication Control)	10.1.0	10.2.0
		SCP(11)0042	218		A	Correction of Contactless state request event description	10.1.0	10.2.0
		SCP(11)0076	219		B	Addition of 'CAT over Modem Interface' when no routing information is provided by application	10.1.0	10.2.0
2011-03	SCP-48	SCP(11)0092r1	220	1	D	Correction of Frame Information Changed event description	10.2.0	10.3.0
		SCP(11)0093	221		D	Correction of Network Search Mode Changed event	10.2.0	10.3.0
		SCP(11)0094	222		F	Correction of default routing for CAT over modem interface	10.2.0	10.3.0
2011-05	SCP-49	SCP(11)0178	223	1	B	Encapsulated CAT commands and envelopes without security	10.3.0	10.4.0
		SCP(11)0179r1	224	1	D	Reservation of values for 3GPP IMS registration event Incoming IMS data envelopes and OPEN CHANNEL for IMS	10.3.0	10.4.0
2011-07	SCP-51	SCP(11)0223r1	225	1	F	CR 102 223 R10 #225r1: Allocation of a Terminal Profile bit for BIP commands for IMS	10.4.0	10.5.0

Change history								
Date	Meeting	TC SCP Doc.	CR	Rv	Cat	Subject/Comment	Old	New
2011-09	SCP-52	SCP(11)0283r2	226	2	F	Corrections for encapsulated CAT	10.5.0	10.6.0
2011-12	SCP-53	SCP(11)0368r1	231	1	A	Removal of obsolete service indication MMS transfer	10.5.0	10.6.0
2011-09	SCP-52	SCP(11)0284r1	227		D	Corrections to Access Technology Change event	10.6.0	11.0.0
2012-09	SCP-56	SCP(12)000147	233		A	Terminal Profile bits for launch parameters and direct application channel	11.0.0	11.1.0
2012-09	SCP-56	SCP(12)000149	234		F	Incomplete command list and class information	11.0.0	11.1.0
2012-09	SCP-56	SCP(12)000145r1	235	1	F	Correction Of References for MEID	11.0.0	11.1.0
2012-09	SCP-56	SCP(12)000148r1	236		C	User Confirmation Clarification	11.0.0	11.1.0
2012-12	SCP-57	SCP(12)000256	237		F	CR 102 223 R11 #237: Incomplete class information in TERMINAL PROFILE	11.1.0	11.2.0
2012-12	SCP-57	SCP(12)000264r1	238	1	B	CR 102 223 R11 #238r1: Security for encapsulated CAT	11.1.0	11.2.0
2012-12	SCP-57	SCP(12)000257r1	239	1	D	CR 102 223 R11 #239r1: Clarification of the Coding of Address	11.1.0	11.2.0
2013-07	SCP-60	SCP(13)000152r1	241	4	F	Securing CAT service lists	11.2.0	11.3.0
2013-07	-	SCP(11)0369	245	-	F	Clarification of service activation for eCAT (CR renumbered to 245 due to duplicate allocation, implemented late for Rel-11 due to lost track)	11.2.0	11.3.0
2014-04	SCP-63	SCP(14)000095	255	-	A	CR 102 223 R11 #255: Correction for eCAT	11.2.0	11.3.0
2014-04	SCP-63	SCP(14)000098	259	-		CR 102 223 R11 #259: Several correction for secured eCAT	11.2.0	11.3.0
2014-04	SCP-63	SCP(14)000099r1	260	1		CR 102 223 R11 #260r1: User confirmation clarification for SET UP CALL command	11.2.0	11.3.0

---

## History

<b>Document history</b>		
V11.0.0	March 2012	Publication
V11.1.0	December 2012	Publication
V11.2.0	July 2013	Publication
V11.3.0	June 2014	Publication