

GSM Protocol Stack



ACI - Application Control Interface **Functional Interface Description** **Confidential**

Author: Condat AG

Date: January 05, 2000

ID: 8411.802.00.105

Status: being processed

Table of Contents

1	Document Control.....	10
1.1	Document History.....	10
1.2	References.....	11
1.3	Abbreviations	11
1.4	Terms.....	11
2	Introduction	12
3	Interfacing with the ACI	13
3.1	Set Commands.....	13
3.2	Query Commands.....	13
3.3	Test Parameter Commands.....	14
3.4	Final Results.....	14
3.5	Unsolicited Messages.....	15
3.6	Return Codes	15
3.7	Parameter Convention.....	16
3.8	Naming Convention	16
4	Data Types.....	17
4.1	General	17
4.2	Simple Type Definitions.....	18
4.2.1	T_ACI_ERR_DESC.....	18
4.3	Enumerations.....	18
4.3.1	T_ACI_AT_CMD	18
4.3.2	T_ACI_BS_FRM.....	19
4.3.3	T_ACI_BS_PAR:.....	19
4.3.4	T_ACI_BS_SPEED.....	20
4.3.5	T_ACI_CAL_IBT.....	20
4.3.6	T_ACI_CALL_MODE.....	20
4.3.7	T_ACI_CAL_MPTY	21
4.3.8	T_ACI_CAL_OWN.....	21
4.3.9	T_ACI_CAL_STAT.....	21
4.3.10	T_ACI_CAL_TYPE.....	22
4.3.11	T_ACI_CBST_CE.....	22
4.3.12	T_ACI_CBST_NAM.....	22
4.3.13	T_ACI_CCFC_MOD.....	22
4.3.14	T_ACI_CCFC_RSN.....	22
4.3.15	T_ACI_CCUG_IDX	23
4.3.16	T_ACI_CCUG_INFO.....	23
4.3.17	T_ACI_CCUG_MOD	23
4.3.18	T_ACI_CCWA_MOD.....	23
4.3.19	T_ACI_CEER.....	24
4.3.20	T_ACI_CFUN_FUN.....	25
4.3.21	T_ACI_CFUN_RST	25
4.3.22	T_ACI_CHLD_MOD	25
4.3.23	T_ACI_CLASS.....	26
4.3.24	T_ACI_CLCC_DIR	26
4.3.25	T_ACI_CLCC_MODE	26
4.3.26	T_ACI_CLCC_MPTY.....	27
4.3.27	T_ACI_CLCC_STAT.....	27
4.3.28	T_ACI_CLCK_FAC.....	27
4.3.29	T_ACI_CLCK_MOD.....	28
4.3.30	T_ACI_CLIP_STAT.....	28
4.3.31	T_ACI_CLIR_MOD	28
4.3.32	T_ACI_CLIR_STAT.....	28

4.3.33	T_ACI_CLOG_TYPE	29
4.3.34	T_ACI_CMD_SRC	29
4.3.35	T_ACI_CME_ERR	29
4.3.36	T_ACI_CMGF_MOD	30
4.3.37	T_ACI_CMOD_MOD	30
4.3.38	T_ACI_CMS_ERR	30
4.3.39	T_ACI_CNMI_MT	32
4.3.40	T_ACI_CNMI_BM	33
4.3.41	T_ACI_CNMI_DS	33
4.3.42	T_ACI_CNUM_ITC	33
4.3.43	T_ACI_CNUM_MOD	33
4.3.44	T_ACI_CNUM_SERV	33
4.3.45	T_ACI_COLP_STAT	34
4.3.46	T_ACI_COLR_STAT	34
4.3.47	T_ACI_COPN_LID	34
4.3.48	T_ACI_COPS_FRMT	34
4.3.49	T_ACI_COPS_MOD	34
4.3.50	T_ACI_COPS_STAT	35
4.3.51	T_ACI_CPAS_PAS	35
4.3.52	T_ACI_CPBF_MOD	35
4.3.53	T_ACI_CPI_IBT	35
4.3.54	T_ACI_CPI_MSG	36
4.3.55	T_ACI_CPI_TCH	36
4.3.56	T_ACI_CPIN_RSLT	36
4.3.57	T_ACI_CPOL_FRMT	36
4.3.58	T_ACI_CPWD_FAC	36
4.3.59	T_ACI_CR_SERV	37
4.3.60	T_ACI_CREG_STAT	37
4.3.61	T_ACI_CRING_MOD	37
4.3.62	T_ACI_CRING_TYP	38
4.3.63	T_ACI_CRSM_CMD	38
4.3.64	T_ACI_CSCB_MOD	38
4.3.65	T_ACI_CSCS_CHSET	38
4.3.66	T_ACI_CSMS_SERV	39
4.3.67	T_ACI_CSMS_SUPP	39
4.3.68	T_ACI_CSNS_MOD	39
4.3.69	T_ACI_CSSI_CODE	39
4.3.70	T_ACI_CSSU_CODE	40
4.3.71	T_ACI_CUSD_MOD	40
4.3.72	T_ACI_D_CLIR_OVRD	40
4.3.73	T_ACI_D_CUG_CTRL	41
4.3.74	T_ACI_D_TOC	41
4.3.75	T_ACI_DR_TYP	41
4.3.76	T_ACI_DS_COMP	41
4.3.77	T_ACI_DS_DIR	41
4.3.78	T_ACI_ERR_CLASS	42
4.3.79	T_ACI_EXT_ERR	42
4.3.80	T_ACI_KSD_CMD	42
4.3.81	T_ACI_NRG_FRMT	43
4.3.82	T_ACI_NRG_RGMD	43
4.3.83	T_ACI_NRG_SVMD	43
4.3.84	T_ACI_PB_STOR	43
4.3.85	T_ACI_PBCF_LDN	44
4.3.86	T_ACI_PBCF_LRN	44

4.3.87	T_ACI_PVRF_STAT	44
4.3.88	T_ACI_PVRF_TYPE	44
4.3.89	T_ACI_RETURN.....	44
4.3.90	T_ACI_RLOG_RSLT.....	45
4.3.91	T_ACI_SATT_CS.....	45
4.3.92	T_ACI_SMS_READ.....	45
4.3.93	T_ACI_SMS_STAT.....	45
4.3.94	T_ACI_SMS_STOR	45
4.3.95	T_ACI_STATUS	46
4.3.96	T_ACI_TOA_NPI	46
4.3.97	T_ACI_TOA_TON.....	46
4.3.98	T_ACI_TOS_OE	46
4.3.99	T_ACI_TOS_TOS.....	46
4.3.100	T_ACI_VTS_MOD.....	47
4.3.101	T_ACI_WS46_MOD.....	47
4.4	Structures.....	48
4.4.1	T_ACI_CAL_ENTR	48
4.4.2	T_ACI_CBM_DATA	48
4.4.3	T_ACI_CCFC_SET.....	48
4.4.4	T_ACI_CLCC_CALDESC	48
4.4.5	T_ACI_CLOG	49
4.4.5.1	cmdPrm.....	49
4.4.6	T_ACI_CLSSTAT.....	58
4.4.7	T_ACI_CMD_DATA	58
4.4.8	T_ACI_CMGL_CBM	59
4.4.9	T_ACI_CMGL_SM	59
4.4.10	T_ACI_CMGR_CBM	59
4.4.11	T_ACI_CMGR_SM	59
4.4.12	T_ACI_CNUM_MSISDN.....	60
4.4.13	T_ACI_COPN_OPDESC.....	60
4.4.14	T_ACI_COPS_OPDESC	60
4.4.15	T_ACI_CPOL_OPDESC	61
4.4.16	T_ACI_KSIR.....	61
4.4.16.1	ir.....	61
4.4.17	T_ACI_PB_ENTR	64
4.4.18	T_ACI_PB_TEXT.....	64
4.4.19	T_ACI_RLOG	64
4.4.19.1	rsltPrm	64
4.4.20	T_ACI_RPCT_VAL.....	66
4.4.21	T_ACI_SM_DATA	66
4.4.22	T_ACI_SMS_STOR_OCC	66
4.4.23	T_ACI_TOA	66
4.4.24	T_ACI_TOS	66
4.4.25	T_ACI_UDH_DATA	66
4.4.26	T_ACI_USSD_DATA	67
4.4.27	T_ACI_VP_ABS.....	67
4.5	Arrays	67
4.5.1	T_ACI_CAL_LST.....	67
4.5.2	T_ACI_CCFC_LST	67
4.5.3	T_ACI_CLCC_LST	67
4.5.4	T_ACI_CLSSTAT_LST.....	67
4.5.5	T_ACI_CMGL_CBM_LST.....	67
4.5.6	T_ACI_CMGL_SM_LST.....	68
4.5.7	T_ACI_CNUM_LST.....	68
4.5.8	T_ACI_COPS_LST	68

4.5.9	T_ACI_PB_LST	68
5	Constants	69
5.1	General	69
5.2	Definitions	69
6	Functions Derived from GSM Rec. 07.07	71
6.1	General Commands	71
6.1.1	Set Functions	71
6.1.1.1	sAT_PlusWS46 – Select Wireless Network	71
6.1.2	Query Functions	72
6.1.2.1	qAT_PlusCIMI () – Request International Mobile Subscriber Identity	72
6.1.2.2	qAT_PlusWS46 – Select Wireless Network	73
6.1.3	Call-back Functions	74
6.1.3.1	rAT_PlusCIMI () – International Mobile Subscriber Identity	74
6.2	Call Control Functions	75
6.2.1	Set Functions	75
6.2.1.1	sAT_PlusCSTA () - Select Type of Address	75
6.2.1.2	sAT_Dn () - Originate Call to Phone Number Provided	76
6.2.1.3	sAT_Dm () - Originate a Call Using Phonebook Memory	77
6.2.1.4	sAT_PlusCMOD () - Call Mode	79
6.2.1.5	sAT_PlusCHUP () - Hangup Call	80
6.2.1.6	sAT_PlusCBST () - Select Bearer Service Type	81
6.2.1.7	sAT_PlusCRLP () Radio Link Protocol	82
6.2.1.8	sAT_PlusCSNS () Single Numbering Scheme	83
6.2.2	Query Functions	84
6.2.2.1	qAT_PlusCSTA () - Select Type of Address	84
6.2.2.2	qAT_PlusCMOD () - Call Mode	85
6.2.2.3	qAT_PlusCBST () - Bearer Service Type Selection	86
6.2.2.4	qAT_PlusCRLP () Radio Link Protocol	87
6.2.2.5	qAT_PlusCEER () – Extended Error Report	88
6.2.2.6	qAT_PlusCSNS () – Single Numbering Scheme	89
6.2.3	Call-back Functions	90
6.2.3.1	rAT_PlusCR () - Service Reporting Control	90
6.2.3.2	rAT_PlusCRING () - Incoming Call Indication	91
6.2.3.3	rAT_PlusCRING_OFF () - Incoming Call Indication OFF	92
6.3	Network Service Related Functions	93
1.	Set Functions	93
6.3.1.1	sAT_PlusCOPS () - Operator Selection	93
6.3.1.2	sAT_PlusCLCK () - Facility Lock	94
6.3.1.3	sAT_PlusCPWD () - Change Password	95
6.3.1.4	sAT_PlusCLIR () - Calling Line Identification Restriction	96
6.3.1.5	sAT_PlusCCUG () - Closed User Group	97
6.3.1.6	sAT_PlusCCFC () - Call Forwarding Number and Conditions	98
6.3.1.7	sAT_PlusCCWA () - Call Waiting	99
6.3.1.8	sAT_PlusCHLD () - Call Hold and Multiparty	100
6.3.1.9	sAT_PlusCTFR () - Call Transfer (not implemented yet)	101
6.3.1.10	sAT_PlusCUSD () – Unstructured Supplementary Service Data	102
6.3.1.11	sAT_PlusCPOL() – Preferred Operator List	103
6.3.2	Query Functions	104
6.3.2.1	qAT_PlusCNUM () - Subscriber Number	104
6.3.2.2	qAT_PlusCREG () - Network Registration Info	105
6.3.2.3	qAT_PlusCOPS () - Operator Selection	106
6.3.2.4	qAT_PlusCLCK () - Facility Lock	107
6.3.2.5	qAT_PlusCLIP () - Calling Line Identification Presentation	108
6.3.2.6	qAT_PlusCLIR () - Calling Line Identification Restriction	109
6.3.2.7	qAT_PlusCOLP () - Connected Line Identification Presentation	110

6.3.2.8	qAT_PlusCCUG () - Closed User Group	111
6.3.2.9	qAT_PlusCCFC () - Call Forwarding Number and Conditions	112
6.3.2.10	qAT_PlusCCWA () - Call Waiting	113
6.3.2.11	qAT_PlusCAOC () - Advice of Charge	114
6.3.2.12	qAT_PlusCLCC () - List Current Calls.....	115
6.3.2.13	qAT_PlusCPOL () - Preferred Operator List	116
6.3.2.14	qAT_PlusCOPN () - Read Operator Names.....	117
6.3.3	Test Functions.....	118
6.3.3.1	tAT_PlusCOPS () - Operator Selection	118
6.3.3.2	tAT_PlusCPOL () - Preferred Operator List	119
6.3.4	Call-back Functions	120
6.3.4.1	rAT_PlusCNUM () - Subscriber Number	120
6.3.4.2	rAT_PlusCOPS () - Operator Selection.....	121
6.3.4.3	rAT_PlusCREG () - Network Registration	122
6.3.4.4	rAT_PlusCLCK () - Facility Lock	123
6.3.4.5	rAT_PlusCLIP () - Calling Line Identification Presentation.....	124
6.3.4.6	rAT_PlusCLIR () - Calling Line Identification Restriction	125
6.3.4.7	rAT_PlusCOLP () - Connected Line Identification Presentation	126
6.3.4.8	rAT_PlusCCFC () - Call Forwarding Number and Conditions	127
6.3.4.9	rAT_PlusCCWA () - Call Waiting.....	128
6.3.4.10	rAT_PlusCUSD () - Unstructured Supplementary Service Data	129
6.3.4.11	rAT_PlusCCCM () - Change Current Call Meter	130
6.3.4.12	rAT_PlusCSSI () - Supplementary Service Notification.....	131
6.3.4.13	rAT_PlusCSSU () - Supplementary Service Notification	132
6.3.4.14	rAT_PlusCPOL () - Preferred Operator List.....	133
6.4	ME Control and Status Functions	134
6.4.1	Set Functions.....	134
6.4.1.1	sAT_PlusCFUN () - Set Phone Functionality.....	134
6.4.1.2	sAT_PlusCPIN () - Enter PIN.....	135
6.4.1.3	sAT_PlusCPBS () - Select Phonebook Memory Storage.....	136
6.4.1.4	sAT_PlusCPBR () - Read Phonebook Entries.....	137
6.4.1.5	sAT_PlusCPBF () - Find Phonebook Entries.....	138
6.4.1.6	sAT_PlusCPBW () - Write Phonebook Entry.....	139
6.4.1.7	sAT_PlusCACM () - Accumulated Call Meter.....	140
6.4.1.8	sAT_PlusCamm () - Accumulated Call Meter Maximum	141
6.4.1.9	sAT_PlusCPUC () - Price per Unit and Currency Table.....	142
6.4.1.10	sAT_PlusCRSM() - Restricted SIM Access	143
6.4.2	Query Functions	144
6.4.2.1	qAT_PlusCPAS () - Phone Activity Status.....	144
6.4.2.2	qAT_PlusCFUN () - Phone Functionality.....	145
6.4.2.3	qAT_PlusCPIN () - Enter PIN	146
6.4.2.4	qAT_PlusCPBS () - Select Phonebook Memory Storage.....	147
6.4.2.5	qAT_PlusCACM () - Accumulated Call Meter	148
6.4.2.6	qAT_PlusCamm () - Accumulated Call Meter Maximum	149
6.4.2.7	qAT_PlusCPUC () - Price per Unit and Currency Table	150
6.4.3	Test Functions.....	151
6.4.3.1	tAT_PlusCPBR () - Read Phonebook Entries	151
6.4.3.2	tAT_PlusCPBF () - Find Phonebook Entries	152
6.4.3.3	tAT_PlusCPBW () - Write Phonebook Entry	153
6.4.4	Call-back Functions	154
6.4.4.1	rAT_PlusCPIN () - Enter PIN	154
6.4.4.2	rAT_PlusCRSM () - Restricted SIM Access	155
6.5	ME Errors	156
6.5.1	Call-back Functions	156
6.5.1.1	rAT_PlusCME () - ME Error Result Code	156
6.6	Functions Diverted from TIA IS-101	157

6.6.1	Set Functions	157
6.6.1.1	sAT_PlusVTS () - DTMF Generation	157
7	Functions Diverted from GSM Rec. 07.05	158
7.1	General Configuration Functions	158
7.1.1	Set Functions	158
7.1.1.1	sAT_PlusCSMS () - Select Message Service	158
7.1.1.2	sAT_PlusCPMS () - Preferred Message Storage	159
7.1.1.3	sAT_PlusCMGF () - SMS Format	160
7.1.2	Query Functions	161
7.1.2.1	qAT_PlusCSMS () - Select Message Service	161
7.1.2.2	qAT_PlusCPMS () - Preferred Message Storage	162
7.1.2.3	qAT_PlusCMGF () - SMS Format	163
7.1.3	Call-back Functions	164
7.1.3.1	rAT_PlusCSMS () - Select Message Service.....	164
7.1.3.2	rAT_PlusCPMS () - Preferred Message Storage.....	165
7.1.3.3	rAT_PlusCMS () - Message Service Failure	166
7.2	Message Configuration Functions	167
7.2.1	Set Functions	167
7.2.1.1	sAT_PlusCSCA () - Service Centre Address	167
7.2.1.2	sAT_PlusCSMP () - Set Text Mode Parameters	168
7.2.1.3	sAT_PlusCSCB () - Select Cell Broadcast Message Types.....	169
7.2.1.4	sAT_PlusCSAS () - Save Settings	170
7.2.1.5	sAT_PlusCRES () - Restore Settings	171
7.2.2	Query Functions	172
7.2.2.1	qAT_PlusCSCA () - Service Centre Address.....	172
7.2.2.2	qAT_PlusCSMP () - Set Text Mode Parameters.....	173
7.2.2.3	qAT_PlusCSCB () - Select Cell Broadcast Message Types	174
7.3	Message Receiving and Reading Functions	175
7.3.1	Set Functions	175
7.3.1.1	sAT_PlusCNMI () - New Message Indications to TE	175
7.3.1.2	sAT_PlusCMGL () - List Messages.....	176
7.3.1.3	sAT_PlusCMGR () - Read Message.....	177
7.3.1.4	sAT_PlusCNMA () - New Message Acknowledgement to ME/TA	178
7.3.2	Query Functions	179
7.3.2.1	qAT_PlusCNMI () - New Message Indications to TE.....	179
7.3.3	Call-back Functions	180
7.3.3.1	rAT_PlusCBM () - New Message Indication for a Cell Broadcasting Message	180
7.3.3.2	rAT_PlusCBMI () - New Message Indication for a Cell Broadcasting Message.....	181
7.3.3.3	rAT_PlusCDS () - New Message Indication for a Status Report.....	182
7.3.3.4	rAT_PlusCMT () - New Message Indication for Mobile Terminated Messages	183
7.3.3.5	rAT_PlusCMTI () - New Message Indication for Mobile Terminated Messages.....	184
7.3.3.6	rAT_PlusCMGL () - List Messages.....	185
7.3.3.7	rAT_PlusCMGR () - Read Messages.....	186
7.4	Message Sending and Writing Functions	187
7.4.1	Set Functions	187
7.4.1.1	sAT_PlusCMGS () - Send Message	187
7.4.1.2	sAT_PlusCMSS () - Send Message from Storage	188
7.4.1.3	sAT_PlusCMGW () - Write Message to Memory.....	189
7.4.1.4	sAT_PlusCMGD () - Delete Message.....	190
7.4.1.5	sAT_PlusCMGC () - Send Command	191
7.4.2	Call-back Functions	192
7.4.2.1	rAT_PlusCMGS () - Send Message	192
7.4.2.2	rAT_PlusCMSS () - Send Message from Storage	193
7.4.2.3	rAT_PlusCMGW () - Write Message to Memory.....	194
7.4.2.4	rAT_PlusCMGC () - Send Command.....	195
7.5	PDU Mode	196

7.5.1	Set Functions	196
7.5.1.1	sAT_PlusCMGSPdu () - Send Message in PDU Mode	196
7.5.1.2	sAT_PlusCMGCPdu () - Send Command in PDU Mode	198
7.5.1.3	sAT_PlusCNMAPdu () - New Message Acknowledgement to ME/TA in PDU Mode	199
8	Functions Diverted from ITU-T V.25ter	201
8.1	Generic TA Control Commands	201
8.1.1	Set Functions	201
8.1.1.1	sAT_Z () - Reset to Default Configuration	201
8.1.1.2	sAT_AndF () - Select Manufacturer Defaults	202
8.1.1.3	sAT_PlusGCI () - Select Country of Installation (not implemented yet)	203
8.1.2	Query Functions	204
8.1.2.1	qAT_PlusGCI () - Select Country of Installation (not implemented yet)	204
8.2	TE-TA Interface Commands	205
8.2.1	Set Functions	205
8.2.1.1	sAT_PlusICF () - TE-TA Character Framing	205
8.2.2	Query Functions	206
8.2.2.1	qAT_PlusICF () - TE-TA Character Framing	206
8.3	Call Control Functions	207
8.3.1	Set Functions	207
8.3.1.1	sAT_A () - Answer a Call.....	207
8.3.1.2	sAT_H () - Hook Control.....	208
8.3.1.3	sAT_O () - Return to Online Mode.....	209
8.3.2	Call-back Functions	210
8.3.2.1	rAT_BUSY () - Busy Signal Detected	210
Command Reference:	210
8.3.2.2	rAT_CONNECT () - Entering Data Transfer State.....	211
8.3.2.3	rAT_NO_ANSWER () - Connection Completion Timeout.....	212
8.3.2.4	rAT_NO_CARRIER () - Connection Terminated	213
8.3.2.5	rAT_OK () - Command Successfully Completed	214
8.4	Data Compression Functions	215
8.4.1	Set Functions	215
8.4.1.1	sAT_PlusDS () - Data Compression	215
8.4.2	Query Functions	216
8.4.2.1	qAT_PlusDS () - Data Compression.....	216
8.4.3	Call-back Functions:	217
8.4.3.1	rAT_PlusDR () Data Compression.....	217
9	Miscellaneous	218
9.1.1	Set Functions	218
9.1.1.1	sAT_Abort () Abort execution for pending command	218
9.1.1.2	sAT_PercentNRG () - Set Registration Mode and Service Mode	219
Command Reference:	219
9.1.1.3	sAT_PercentPBCF () - Set Phone Configuration	220
Command Reference:	220
9.1.1.4	sAT_PercentPVRF () - Verify a Specific Pin	221
9.1.1.5	sAT_PercentSATC () - Configure SIM Toolkit Profile	222
9.1.1.6	sAT_PercentSATE () - Envelope SIM Toolkit Command	223
9.1.1.7	sAT_PercentSATR () - Response on STK Command.....	224
9.1.1.8	sAT_PercentSATT () - Sim Application Toolkit Termination.....	225
9.1.2	Query Functions	226
9.1.2.1	qAT_ErrDesc () - Query Error Description of last Error	226
9.1.2.2	qAT_PercentCAL () - Query Current Call Status.....	228
9.1.2.3	qAT_PercentCAOC () - Query Current Call Meter using PUCT.....	229
9.1.2.4	qAT_PercentCOLR () - Query Setting for Connected Line Restriction Supplementary Service	230
9.1.2.5	qAT_PercentCTV () - Query Current Call Timer.....	231
9.1.2.6	qAT_PercentNRG () - Query Current Registration Settings.....	232

9.1.2.7	qAT_PercentPBCF () – Query Phone Configuration.....	233
9.1.2.8	qAT_PercentPVRF () - Query the Current Counter for PIN and PUK	234
9.1.2.9	qAT_PercentRPCT () - Query Raw SIM Data for PUCT	235
9.1.3	Call-back Functions	236
9.1.3.1	rAT_PercentCLOG () – Command Logging	236
9.1.3.2	rAT_PercentCOLR () - Connected Line Identification Restriction	237
9.1.3.3	rAT_PercentCPI () – Call Progress Information	238
9.1.3.4	rAT_PercentCTV () – Call Timer	239
9.1.3.5	rAT_PercentKSIR () – Keystroke Interaction Result	240
9.1.3.6	rAT_PercentRLOG () – Result Logging	241
9.1.3.7	rAT_PercentSATA () – Alert about Sim Application Toolkit Pending Call Setup	242
9.1.3.8	rAT_PercentSATE () – Notify Application about Result of Envelope Command	243
9.1.3.9	rAT_PercentSATI () – Indicate SAT Command Sent by SIM.....	244
9.1.3.10	rAT_PercentSATN () – Notify Application while Call Setup or Supplementary Service	245
9.1.3.11	rAT_PercentSIMINS () – SIM Inserted	246
9.1.3.12	rAT_PercentSIMREM () – SIM Removed.....	247
10	Example	248

1 Document Control

© Copyright AG, 2000.

All rights reserved.

Every effort has been made to ensure that the information contained in this document is accurate at the time of printing. However, the software described in this document is subject to continuous development and improvement. Condat AG reserves the right to change the specification of the software. Information in this document is subject to change without notice and does not represent a commitment on the part of Condat AG. Condat AG accepts no liability for any loss or damage arising from the use of any information contained in this document.

The software described in this document is furnished under a license agreement and may be used or copied only in accordance with the terms of the agreement. It is an offence to copy the software in any way except as specifically set out in the agreement. No part of this document may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, for any purpose without the express written permission of Condat AG.

Condat AG
Alt Moabit 91d
10559 Berlin
Germany

Telephone: +49.30.39094-0
Fax: +49.30.39094-300
Internet: www.condat.de
E-mail: gsm@condat.de

1.1 Document History

ID	Author	Date	Approval	Remarks
8411.802.98.100	AK, SAB et al.	Sep. 17, 98		Initial, Fax & Data not complete
8411.802.98.101	MS et al.	Dec. 08, 98		Format/English check
8411.802.98.102	SAB et al.	Dec. 21, 98		Adding new functions, types and constants. Fax & Data in separate document.
8411.802.98.103	UB, et al.	Jan. 22, 98		New WCS template
8411.802.00.10	MS, et al.	Jan. 29, 99		Format/English check
8411.802.00.105	AK, HM et al.	Jan. 05, 00		Revision of functions, types and constants.

1.2 References

- [GSM_07.05] GTS 07.05: January 1998 (GSM 07.05 version 5.5.0)
Use of Data Terminal Equipment - Data Circuit terminating Equipment (DTE - DCE) interface for Short Message Service (SMS) and Cell Broadcast Service (CBS); ETSI
- [GSM_07.07] ETS 300 916: February 1998 (GSM 07.07 version 5.5.0)
AT command set for GSM Mobile Equipment (ME); ETSI
- [T.32] T.32 (08/95)
Asynchronous facsimile DCE control - service class 2; ITU
- [T.25ter] T V.25 ter, 1997
ITU-T Recommendation V.25 ter; Series V: data communication over the telephone network; Interfaces and voiceband modems; Serial asynchronous automatic dialing and control; ITU
- [C_8410.011] 8410.011.99.102; January 21, 1999
ACI - Application Control Interface, AT Command Interface Description; Condat
- [C_8415.022] 8415.022.99.001; January 22, 1999
ACI - Application Control Interface, Functional Interface Description Part 2: F&D; Condat

1.3 Abbreviations

ACI	Application Control Interface (AT Commands)
DTE	Data Terminal Equipment
MMI	Man Machine Interface
SAT	Sim Application Toolkit

1.4 Terms

2 Introduction

G23 is a software package implementing Layers 2 and 3 of the ETSI-defined GSM air interface signalling protocol, and as such represents the part of a GSM mobile station's protocol software which is both, platform and manufacturer independent. Therefore, G23 can be viewed as a building block providing standardized functionality through generic interfaces for easy integration.

The G23 suite of products consists of the following items:

- Layers 2 and 3 for speech & short message services,
- Layers 2 and 3 for fax & data services,
- Application Control Interface,
- Slim MMI [02.30] and
- Test and integration support tools.

The ACI supports a functional interface which allows a target MMI to access the ACI entity most directly. This interface is based on the AT Command sets described by the specifications GSM 7.07 and 7.05, V.25ter and CCITT-T.32 for Class 2.0 FAX modems. For each AT Command that must be executed, the appropriate function for that command must be called by the application. Depending on the command type, an action is performed by the function or requested parameters are returned. This is done by exchanging primitive interactions with Layer 3 of the protocol stack. Requested parameters or result codes recognized instantly by the ACI are returned immediately by the functions, whereas parameters which must be requested from the protocol stack are passed using a call-back mechanism as soon as they are available.

3 Interfacing with the ACI

The following describes the conventions used for transforming the ASCII-based AT Commands into the functional AT interface.

3.1 Set Commands

For each AT Command that sets a functionality of the ACI, a corresponding function is provided by the Command Handler. The function requires that a parameter is passed for each AT Command parameter expected. Optional AT Command parameters which are not presented by the AT Command must be indicated as invalid (e.g. -1 for integer parameters and null for string pointers). This set of functions has the names prefixed by the character 's' (set).

Example:

	AT Command Interface	Functional Interface
declaration	AT+CPIN=<pin>[,<newpin>]	T_ACI_RETURN sAT_PlusCPIN(T_ACI_CMD_SRC srcId, CHAR* pin, CHAR* newpin);
use	AT+CPIN="1234"	CHAR* pPIN = "1234"; sAT_PlusCPIN(CMD_SRC_RMT, pPIN, NULL);

3.2 Query Commands

For each AT Command that queries a setting of the ACI, a corresponding function is provided by the Command Handler. The function requires that a reference is passed to a buffer for each parameter that can be queried by the AT Command. The function copies the current settings for the parameters into the passed buffers. Optional parameters which can not be provided by the function are indicated as invalid (e.g. -1 for integer parameters and null for string pointers). This set of functions has the names prefixed by the character 'q' (query).

Example:

	AT Command Interface	Functional Interface
declaration	AT+CLIP?	T_ACI_RETURN qAT_PlusCLIP(T_ACI_CMD_SRC srcId, T_ACI_CLIP_STAT* stat);
use	AT+CLIP?	T_ACI_CLIP_STAT curStat; qAT_PlusCLIP(CMD_SRC_LCL, &curStat);

3.3 Test Parameter Commands

For each AT Command that tests the ACI for possible parameters and their setting ranges, **no** function is provided by the Command Handler. It is assumed that the use of the functional interface is recognized by the local application on top of the ACI and that this information does not need be queried. These commands are sensible for a remote application which has no knowledge of the AT Commands supported. The only exceptions are commands where a test requests volatile information which cannot be foreseen by the application (e.g. the list of available PLMNs). The interface of that function requires that a reference is passed to a buffer for each parameter which can be queried by the AT Command. The function copies the current settings for the parameters into the passed buffers. Optional parameters which can not be provided by the function are indicated as invalid (e.g. -1 for integer parameters and null for string pointers). This set of functions has the names prefixed by the character 't' (**t**est).

Example:

	AT Command Interface	Functional Interface
declaration	AT+COPS=?	T_ACI_RETURN tAT_PlusCOPS(T_ACI_CMD_SRC srcId, SHORT startIdx, SHORT* lastIdx, T_ACI_COPS_LST* operLst);
use	AT+COPS=?	T_ACI_COPS_LST operLst[MAX_OPER]; SHORT lastIdx; tAT_PlusCOPS(CMD_SRC_LCL, 0, &lastIdx, operLst);

3.4 Final Results

For each final result of an AT Command, the calling application must provide a function which utilizes a call-back mechanism. If the ACI detects a final result condition (e.g. OK or ERROR), it makes a call to the respective call-back function and passes the necessary parameters. This set of functions has the names prefixed by the character 'r' (response).

Example:

	AT Command Interface	Functional Interface
declaration	+CME ERROR: <err>	void rAT_PlusCME(T_ACI_AT_CMD cmdId, T_ACI_CME_ERR err);
use	AT+CPIN="1234" +CME ERROR: incorrect password	rAT_PlusCME(AT_CMD_CPIN, CME_ERR_IncorPasswd);

3.5 Unsolicited Messages

For each unsolicited message that can be produced by the ACI, the calling application must provide a function which utilizes a call-back mechanism. If the ACI detects a condition that must be indicated (e.g. +CRING or +CLIP:), it makes a call to the respective call-back function and passes the necessary parameters. This set of functions has the names prefixed by the character 'r' (response).

Example:

	AT Command Interface	Functional Interface
declaration	+CRING: <type>	void rAT_PlusCRING(T_ACI_CRING_MOD mode, T_ACI_CRING_TYP type1, T_ACI_CRING_TYP type2);
use	+CRING: VOICE	rAT_PlusCRING(CRING_MOD_Direct, CRING_TYP_Voice, CRING_TYP_NotPresent);

3.6 Return Codes

The return codes of the sAT_, qAT_ and tAT_ functions indicate the command execution state (failed, completed, busy and executing). A function invoked using incorrect function parameters is terminated by returning a fail condition. Functions that do not directly interact with the protocol stack (e.g. the setting of parameters for later use) terminate by returning that the command was completed. Functions which interact directly with the protocol stack (e.g. register with network) return a value which indicates that the command is executing. In this case, the result of the command is passed via the call-back mechanism as soon as the information is available. If the invocation of a command interferes with the execution of a previous command, the function terminates by returning a busy condition. In that case the invocation of that command has to be postponed until the current command execution is terminated.

Example:

	AT Command Interface	Functional Interface
fail	AT+CMOD=5 ERROR	sAT_PlusCMOD(CMD_SRC_RMT, 5) { if(...) /* if input parameters out of range */ return(AT_FAIL); }
complete	AT+CBST=7,0,1 OK	sAT_PlusCBST(CMD_SRC_LCL, 7,0,1) { speed = 7; /* store parameter for later use */ service = 0; ce = 1; return(AT_CMPL); }
executing	AT+CPIN?	qAT_PlusCPIN(CMD_SRC_RMT, pPINbuf) { if(...) /* if information not present */ { sim_activate_req(); /* request information */ }

		<pre> return(AT_EXCT); } } </pre>
	+CPIN: SIM PIN OK	<pre> rAT_PlusCPIN(PINstat); rAT_OK(AT_CMD_CPIN); </pre>

3.7 Parameter Convention

Numerical parameters and the return values of the functional interface are declared as type definitions by the use of enumeration. Therefore, each parameter of an AT Command represents an enumeration type containing all of the supported settings for that parameter.

String parameters of AT Commands contain only ASCII coded characters are passed by a reference, which points to the location of the string. The type of the pointer is CHAR and the string must be terminated by a null character to indicate the end of the string.

String parameters of AT Commands which are coded in GSM default alphabet are passed by a reference, which points to the location of the string. The type of the pointer is UBYTE. An additional parameter indicates the length of the string in bytes, because a null termination interferes with the GSM alphabet character '@'.

The previously mentioned examples provide an overview of how to exchange parameters with the functional interface.

3.8 Naming Convention

The naming convention for the functional interface is defined as follows:

All function names start with the previously described prefixes which indicate the type of the function. This prefix is followed by the fixed sequence `AT_` which is then followed by the AT Command name. The command name is printed in upper case characters except for when it contains special characters such as `+,%,>,&/:_`. These are printed in lower case characters with the first character capitalized.

Example:

	AT command interface	functional interface
naming	AT+CPOL	qAT_PlusCPOL

4 Data Types

4.1 General

This section describes the data types needed to drive the functional interface of the ACI. The items are sorted alphabetically within each chapter.

There are three kinds of tables containing all of the information related to the specific data types. The following summary explains the individual columns in detail:

column header	description
enum type	name of the enumeration type
value	numerical values representing each individual enumerator
symbolic constant	names of the enumerators
comment	description of the purpose of each individual enumerator
column header	description
struct type	name of the structure type
element	names of all structure members
element type	data types of all structure members
comment	description of the purpose of each structure member
column header	description
array type	name of the array type (at this time only one dimensional arrays are supported)
element type	data type of the array elements
element count	number of elements in the array
comment	description of the purpose of the array type

4.2 Simple Type Definitions

4.2.1 T_ACI_ERR_DESC

Type Definition:

defined type	original type	comment
T_ACI_ERR_DESC	ULONG	supplemental error description

4.3 Enumerations

4.3.1 T_ACI_AT_CMD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_AT_CMD	-1	AT_CMD_NotPresent	parameter not present
	0	AT_CMD_NONE	no ACI command identifier
	1	AT_CMD_CFUN	+CFUN command identifier
	2	AT_CMD_CREG	+CREG command identifier
	3	AT_CMD_COPS	+COPS command identifier
	4	AT_CMD_CPOL	+CPOL command identifier
	5	AT_CMD_D	D command identifier
	6	AT_CMD_A	A command identifier
	7	AT_CMD_H	H command identifier
	8	AT_CMD_CLIR	+CLIR command identifier
	9	AT_CMD_CLIP	+CLIP command identifier
	10	AT_CMD_COLP	+COLP command identifier
	11	AT_CMD_CCFC	+CCFC command identifier
	12	AT_CMD_CLCK	+CLCK command identifier
	13	AT_CMD_CCWA	+CCWA command identifier
	14	AT_CMD_CPWD	+CPWD command identifier
	15	AT_CMD_CHUP	+CHUP command identifier
	16	AT_CMD_CHLD	+CHLD command identifier
	17	AT_CMD_CBST	+CBST command identifier
	18	AT_CMD_CR	+CR command identifier
	19	AT_CMD_CMOD	+CMOD command identifier
	20	AT_CMD_CPIN	+CPIN command identifier
	21	AT_CMD_CMGC	+CMGC command identifier
	22	AT_CMD_CNMI	+CNMI command identifier
	23	AT_CMD_CSMS	+CSMS command identifier
	24	AT_CMD_CMGD	+CMGD command identifier
	25	AT_CMD_CMGR	+CMGR command identifier
	26	AT_CMD_CMGW	+CMGW command identifier

enum type	value	symbolic constant	comment
	27	AT_CMD_CMGS	+CMGS command identifier
	28	AT_CMD_CMSS	+CMSS command identifier
	29	AT_CMD_CPMS	+CPMS command identifier
	30	AT_CMD_CSMP	+CSMP command identifier
	31	AT_CMD_CSCA	+CSCA command identifier
	32	AT_CMD_CRES	+CRES command identifier
	33	AT_CMD_CSAS	+CSAS command identifier
	34	AT_CMD_CMGL	+CMGL command identifier
	35	AT_CMD_SQ	%SQ command identifier
	36	AT_CMD_BC	%BC command identifier
	37	AT_CMD_VTS	+VTS command identifier
	38	AT_CMD_NRG	%NRG command identifier
	39	AT_CMD_COLR	%COLR command identifier
	40	AT_CMD_CUSD	+CUSD command identifier
	41	AT_CMD_ABRT	abort command identifier
	42	AT_CMD_CACM	+CACM command identifier
	43	AT_CMD_CAMM	+CAMM command identifier
	44	AT_CMD_CPUC	+CPUC command identifier
	45	AT_CMD_CIMI	+CIMI command identifier
	46	AT_CMD_CNUM	+CNUM command identifier
	47	AT_CMD_SATE	%SATE command identifier
	48	AT_CMD_PVRF	%PVRF command identifier

4.3.2 T_ACI_BS_FRM

Type Declaration:

enum type	Value	symbolic constant	comment
T_ACI_BS_FRM	-1	BS_FRM_NotPresent	parameter not present
	0	BS_FRM_Dat8_Par0_St2	8 data bits, no parity, 2 stop bits
	1	BS_FRM_Dat8_Par1_St1	8 data bits, parity, 1 stop bit
	2	BS_FRM_Dat8_Par0_St1	8 data bits, no parity, 1 stop bit
	3	BS_FRM_Dat7_Par0_St2	7 data bits, no parity, 2 stop bits
	4	BS_FRM_Dat7_Par1_St1	7 data bits, parity, 1 stop bit
	5	BS_FRM_Dat7_Par0_St1	7 data bits, no parity, 1 stop bit

4.3.3 T_ACI_BS_PAR:

Type Declaration:

enum type	Value	symbolic constant	comment
T_ACI_BS_PAR	-1	BS_PAR_NotPresent	parameter not present
	0	BS_PAR_Odd	parity odd
	1	BS_PAR_Even	parity even

enum type	Value	symbolic constant	comment
	2	BS_PAR_Mark	parity mark
	3	BS_PAR_Space	parity space

4.3.4 T_ACI_BS_SPEED

Type Declaration:

enum type	Value	symbolic constant	comment
T_ACI_BS_SPEED	-1	BS_SPEED_NotPresent	parameter not present
	0	BS_SPEED_Auto	autobauding (automatic selection of the speed)
	1	BS_SPEED_300_V21	300 bps (V.21)
	2	BS_SPEED_1200_V22	1200 bps (V.22)
	3	BS_SPEED_1200_75_V23	1200/75 bps (V.23)
	4	BS_SPEED_2400_V22bis	2400 bps (V.22bis)
	5	BS_SPEED_2400_V26ter	2400 bps (V.26ter)
	6	BS_SPEED_4800_V32	4800 bps (V.32)
	7	BS_SPEED_9600_V32	9600 bps (V.32)
	12	BS_SPEED_9600_V34	9600 bps (V.34)
	14	BS_SPEED_14400_V34	14400 bps (V.34)
	34	BS_SPEED_1200_V120	1200 bps (V.120)
	36	BS_SPEED_2400_V120	2400 bps (V.120)
	38	BS_SPEED_4800_V120	4800 bps (V.120)
	39	BS_SPEED_9600_V120	9600 bps (V.120)
	43	BS_SPEED_14400_V120	14400 bps (V.120)
	65	BS_SPEED_300_V110	300 bps (V.110)
	66	BS_SPEED_1200_V110	1200 bps (V.110)
	68	BS_SPEED_2400_V110	2400 bps (V.110)
	70	BS_SPEED_4800_V110	4800 bps (V.110)
	71	BS_SPEED_9600_V110	9600 bps (V.110)
	75	BS_SPEED_14400_V110	14400 bps (V.110)

4.3.5 T_ACI_CAL_IBT

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CAL_IBT	-1	CAL_IBT_NotPresent	parameter not present
	0	CAL_IBT_FALSE	in-band signalling false
	1	CAL_IBT_TRUE	in-band signalling true

4.3.6 T_ACI_CALL_MODE

Type Declaration:

enum type	value	symbolic constant	comment
-----------	-------	-------------------	---------

enum type	value	symbolic constant	comment
T_ACI_CAL_MODE	-1	CAL_MODE_NotPresent	parameter not present
	0	CAL_MODE_Voice	voice mode
	1	CAL_MODE_Data	data mode
	2	CAL_MODE_Fax	fax mode
	3	CAL_MODE_VFD_Voice	voice followed data, voice mode
	4	CAL_MODE_VAD_Voice	voice alternating data, voice mode
	5	CAL_MODE_VAF_Voice	voice alternating fax, voice mode
	6	CAL_MODE_VFD_Data	voice followed data, voice mode
	7	CAL_MODE_VAD_Data	voice alternating data, voice mode
	8	CAL_MODE_VAF_Fax	voice alternating fax, voice mode
	9	CAL_MODE_Unknown	unknown call mode

4.3.7 T_ACI_CAL_MPTY

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CAL_MPTY	-1	CAL_MPTY_NotPresent	parameter not present
	0	CAL_MPTY_NoMember	no member
	1	CAL_MPTY_IsMember	is member

4.3.8 T_ACI_CAL_OWN

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CAL_OWN	-1	CAL_OWN_NotPresent	parameter not present
	0	CAL_OWN_LCL	local call
	1	CAL_OWN_RMT	remote call
	2	CAL_OWN_NONE	no call owner

4.3.9 T_ACI_CAL_STAT

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CAL_STAT	-1	CAL_STAT_NotPresent	parameter not present
	0	CAL_STAT_Held	call held
	1	CAL_STAT_Active	call active
	2	CAL_STAT_Wait	call waiting
	3	CAL_STAT_Dial	call dialing
	4	CAL_STAT_DeactiveReq	call disconnection
	5	CAL_STAT_Incomming	call incoming
	6	CAL_STAT_Alerting	call alerting

4.3.10 T_ACI_CAL_TYPE**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CAL_TYPE	-1	CAL_TYPE_NotPresent	parameter not present
	0	CAL_TYPE_MOC	mobile originated call
	1	CAL_TYPE_MTC	mobile terminated call

4.3.11 T_ACI_CBST_CE**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CBST_CE	-1	CBST_CE_NotPresent	parameter not present
	0	CBST_CE_Transparent	transparent
	1	CBST_CE_NonTransparent	non-transparent
	2	CBST_CE_BothTransPref	both, transparent preferred
	3	CBST_CE_BothNonTransPref	both, non-transparent preferred

4.3.12 T_ACI_CBST_NAM**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CBST_NAM	-1	CBST_NAM_NotPresent	parameter not present
	0	CBST_NAM_Asynch	asynchronous modem
	1	CBST_NAM_Synch	synchronous modem

4.3.13 T_ACI_CCFC_MOD**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CCFC_MOD	-1	CCFC_MOD_NotPresent	parameter not present
	0	CCFC_MOD_Disable	disable
	1	CCFC_MOD_Enable	enable
	3	CCFC_MOD_Register	registration
	4	CCFC_MOD_Erasure	erasure

4.3.14 T_ACI_CCFC_RSN**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CCFC_RSN	-1	CCFC_RSN_NotPresent	parameter not present
	0	CCFC_RSN_Uncond	unconditional
	1	CCFC_RSN_Busy	mobile busy
	2	CCFC_RSN_NoReply	no reply
	3	CCFC_RSN_NotReach	not reachable
	4	CCFC_RSN_Forward	all call forwarding

enum type	value	symbolic constant	comment
	5	CCFC_RSN_CondForward	all conditional call forwarding

4.3.15 T_ACI_CCUG_IDX

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CCUG_IDX	-1	CCUG_IDX_NotPresent	parameter not present
	0	CCUG_IDX_0	CUG index
	1	CCUG_IDX_1	CUG index
	2	CCUG_IDX_2	CUG index
	3	CCUG_IDX_3	CUG index
	4	CCUG_IDX_4	CUG index
	5	CCUG_IDX_5	CUG index
	6	CCUG_IDX_6	CUG index
	7	CCUG_IDX_7	CUG index
	8	CCUG_IDX_8	CUG index
	9	CCUG_IDX_9	CUG index
	10	CCUG_IDX_No	no index

4.3.16 T_ACI_CCUG_INFO

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CCUG_INFO	-1	CCUG_INFO_NotPresent	parameter not present
	0	CCUG_INFO_No	no information
	1	CCUG_INFO_SuppOa	suppress OA
	2	CCUG_INFO_SuppPrefCug	suppress preferential CUG
	3	CCUG_INFO_SuppBoth	suppress OA and preferential CUG

4.3.17 T_ACI_CCUG_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CCUG_MOD	-1	CCUG_MOD_NotPresent	parameter not present
	0	CCUG_MOD_DisableTmp	disable CUG temporary mode
	1	CCUG_MOD_EnableTmp	enable CUG temporary mode

4.3.18 T_ACI_CCWA_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CCWA_MOD	-1	CCWA_MOD_NotInterrogate	network is not interrogated
	0	CCWA_MOD_Disable	disable call waiting
	1	CCWA_MOD_Enable	enable call waiting

4.3.19 T_ACI_CEER**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CEER	-1	CEER_NotPresent	no error
	1	CEER_Unassign	unassigned number
	3	CEER_NoRoute	no route to destination
	6	CEER_ChUnaccept	channel unacceptable
	8	CEER_Barred	operator determined barring
	16	CEER_CallClear	normal call clearing
	17	CEER_UserBusy	user busy
	18	CEER_NoResponse	no user responding
	19	CEER_AlertNoAnswer	user alerting, no answer
	21	CEER_CallReject	call rejected
	22	CEER_NumChanged	number changed
	26	CEER_UserClear	non selected user clearing
	27	CEER_DestOutOfOrder	destination out of order
	28	CEER_NumFormat	invalid number format
	29	CEER_FacilityReject	facility rejected
	30	CEER_StatusEnquiry	response to status enquiry
	31	CEER_Unspecified	normal, unspecified
	34	CEER_NoChanAvail	no channel available
	38	CEER_NetworkOutOfOrder	network out of order
	41	CEER_TempFailure	temporary failure
	42	CEER_SwitchCongest	switching equipment congestion
	43	CEER_InfoDiscard	access information discarded
	44	CEER_ReqChanUnavail	requested channel unavailable
	47	CEER_ResourceUnavail	resources unavailable
	49	CEER_QOS_Unavail	quality of service unavailable
	50	CEER_FAC_Unsubscr	requested facility unsubscribed
	55	CEER_BarredInCUG	incoming calls barred within CUG
	57	CEER_BearerCapNotAuth	bearer capability not authorized
	58	CEER_BearerCapUnavail	bearer capability not available
	63	CEER_ServUnavail	service not available
	65	CEER_BearerNotImpl	bearer service not implemented
	68	CEER_ACM_Max	ACM reached ACM maximum
	69	CEER_FAC_NotImpl	facility not implemented
	70	CEER_RestrictBearerCap	only restricted bearer cap. avail.
	79	CEER_ServNotImpl	service not implemented
	81	CEER_InvalidTI	invalid TI
	87	CEER_UserNotInCUG	no member of CUG

enum type	value	symbolic constant	comment
	88	CEER_IncompatDest	incompatible destination
	91	CEER_InvalidTransNet	invalid transit network selection
	95	CEER_IncorrMsg	incorrect message
	96	CEER_InvalidMandInfo	invalid mandatory information
	97	CEER_MsgTypeNotImpl	message type not implemented
	98	CEER_MsgTypeIncomp	message type incompatible
	99	CEER_InfoElementNotImpl	info element not implemented
	100	CEER_CondInfoElem	conditional info element error
	101	CEER_MsgIncom	message incompatible
	102	CEER_Timer	recovery on time expiry
	111	CEER_Protocol	protocol error
	127	CEER_Interworking	interworking error
	200	CEER_RegBearerServNotAvail	bearer service not available
	201	CEER_NoTransIdAvail	no TI available
	202	CEER_Timer303	timer 303 expiry
	203	CEER_EstabFail	establishment failure
	210	CEER_NoError	no error
	211	CEER_Failed	operation failed
	212	CEER_Timeout	timeout
	213	CEER_BearerServNotCompat	bearer service not compatible

4.3.20 T_ACI_CFUN_FUN

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CFUN_FUN	-1	CFUN_FUN_NotPresent	parameter not present
	0	CFUN_FUN_Minimum	minimum functionality
	1	CFUN_FUN_Full	full functionality

4.3.21 T_ACI_CFUN_RST

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CFUN_RST	-1	CFUN_RST_NotPresent	parameter not present
	0	CFUN_RST_NoReset	do not reset the ME before setting it to <fun> power level. This is always default when <rst> is not provided

4.3.22 T_ACI_CHLD_MOD

Type declaration:

enum type	value	symbolic constant	comment
T_ACI_CHLD_MOD	-1	CHLD_MOD_NotPresent	parameter not present

enum type	value	symbolic constant	comment
	0	CHLD_MOD_RelHldOrUdub	release held call or user defined user busy
	1	CHLD_MOD_RelActAndAcpt	release all active calls and accepts the other held or waiting call
	2	CHLD_MOD_RelActSpec	release active specific
	3	CHLD_MOD_HldActAndAcpt	places all active calls on hold and accept the other held or waiting call
	4	CHLD_MOD_HldActExc	Places all active calls on hold except call X with which communication shall be supported
	5	CHLD_MOD_AddHld	adds a held call to conversation
	6	CHLD_MOD_Ect	explicit call transfer
	7	CHLD_MOD_Redirect	for internal use only
	8	CHLD_MOD_CcbsOrCcnry	Completion of calls to busy subscriber or completion of call to subscriber not reachable (not implemented yet)

4.3.23 T_ACI_CLASS

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CLASS	-1	CLASS_NotPresent	parameter not present
	0	CLASS_None	no class
	1	CLASS_Vce	voice only
	2	CLASS_Dat	data only
	3	CLASS_VceDat	voice and data
	4	CLASS_Fax	fax only
	5	CLASS_VceFax	voice and fax
	6	CLASS_DatFax	data and fax
	7	CLASS_VceDatFax	voice, data and fax

4.3.24 T_ACI_CLCC_DIR

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CLCC_DIR	-1	CLCC_DIR_NotPresent	parameter not present
	0	CLCC_DIR_MOC	mobile originated (MO) call
	1	CLCC_DIR_MTC	mobile terminated (MT) call

4.3.25 T_ACI_CLCC_MODE

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CLCC_MODE	-1	CLCC_MODE_NotPresent	parameter not present
	0	CLCC_MODE_Voice	voice
	1	CLCC_MODE_Data	data

enum type	value	symbolic constant	comment
	2	CLCC_MODE_Fax	fax
	3	CLCC_MODE_VFDVoice	voice followed by data, voice mode
	4	CLCC_MODE_VADVoice	alternating voice/data, voice mode
	5	CLCC_MODE_VAFVoice	alternating voice/fax, voice mode
	6	CLCC_MODE_VFDData	voice followed by data, data mode
	7	CLCC_MODE_VADDData	alternating voice/data, data mode
	8	CLCC_MODE_VAFFax	alternating voice/fax, fax mode
	9	CLCC_MODE_Unknown	unknown

4.3.26 T_ACI_CLCC_MPTY**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CLCC_MPTY	-1	CLCC_MPTY_NotPresent	parameter not present
	0	CLCC_MPTY_NoMember	no member
	1	CLCC_MPTY_IsMember	is member

4.3.27 T_ACI_CLCC_STAT**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CLCC_STAT	-1	CLCC_STAT_NotPresent	parameter not present
	0	CLCC_STAT_Active	active
	1	CLCC_STAT_Held	held
	2	CLCC_STAT_Dialing	dialing (MO call)
	3	CLCC_STAT_Alerting	alerting (MO call)
	4	CLCC_STAT_Incoming	incoming (MT call)
	5	CLCC_STAT_Waiting	waiting (MT call)

4.3.28 T_ACI_CLK_FAC**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CLK_FAC	-1	CLK_FAC_NotPresent	parameter not present
	0	CLK_FAC_Sc	SIM (lock SIM card) (SIM asks for password in ME power-up and when this lock command is issued)
	1	CLK_FAC_Ao	BAOC (Barr All Outgoing Calls) (refer GSM 02.88 [6] clause 1)
	2	CLK_FAC_Oi	BOIC (Barr Outgoing International Calls) (refer to GSM 02.88 [6] clause 1)
	3	CLK_FAC_Ox	BOIC-exHC (Barr Outgoing International Calls except to Home Country) (refer to GSM 02.88 [6] clause 1)
	4	CLK_FAC_Ai	BAIC (Barr All Incoming Calls) (refer to

enum type	value	symbolic constant	comment
			GSM 02.88 [6] clause 2)
	5	CLCK_FAC_Ir	BIC-Roam (Barr Incoming Calls when Roaming outside the home country) (refer to GSM 02.88 [6] clause 2)
	6	CLCK_FAC_Ab	All Barring services (refer to GSM 02.30 [19])
	7	CLCK_FAC_Ag	All outGoing barring services (refer to GSM 02.30 [19])
	8	CLCK_FAC_Ac	All inComing barring services (refer to GSM 02.30 [19])
	9	CLCK_FAC_Fd	SIM fixed dialing memory feature (if PIN2 authentication has not been performed during the current session, PIN2 is required as <passwd>)

4.3.29 T_ACI_CLK_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CLK_MOD	-1	CLCK_MOD_NotPresent	parameter not present
	0	CLCK_MOD_Unlock	unlock
	1	CLCK_MOD_Lock	lock

4.3.30 T_ACI_CLIP_STAT

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CLIP_STAT	-1	CLIP_STAT_NotPresent	parameter not present
	0	CLIP_STAT_NotProv	CLIP not provisioned
	1	CLIP_STAT_Prov	CLIP provisioned
	2	CLIP_STAT_Unknown	unknown (e.g. no network, etc.)

4.3.31 T_ACI_CLIR_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CLIR_MOD	-1	CLIR_MOD_NotPresent	parameter not present
	0	CLIR_MOD_Subscript	presentation indicator is used according to the subscription of the CLIR service
	1	CLIR_MOD_Invoc	CLIR invocation
	2	CLIR_MOD_Supp	CLIR suppression

4.3.32 T_ACI_CLIR_STAT

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CLIR_STAT	-1	CLIR_STAT_NotPresent	parameter not present

enum type	value	symbolic constant	comment
	0	CLIR_STAT_NotProv	CLIR not provisioned
	1	CLIR_STAT_Permanent	CLIR provisioned in permanent mode
	2	CLIR_STAT_Unknown	unknown (e.g. no network, etc.)
	3	CLIR_STAT_RestrictTemp	CLIR temporary mode presentation restricted
	4	CLIR_STAT_AllowTemp	CLIR temporary mode presentation allowed

4.3.33 T_ACI_CLOG_TYPE

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CLOG_TYPE	-1	CLOG_TYPE_NotPresent	Parameter not present
	0	CLOG_TYPE_Set	set command
	1	CLOG_TYPE_Query	query command
	2	CLOG_TYPE_Test	test command

4.3.34 T_ACI_CMD_SRC

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CMD_SRC	-1	CMD_SRC_NONE	no ACI command source identifier
	0	CMD_SRC_LCL	local command source identifier
	1	CMD_SRC_RMT	remote command source identifier
	2	CMD_SRC_MAX	maximum command source identifier

4.3.35 T_ACI_CME_ERR

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CME_ERR	-1	CME_ERR_NotPresent	parameter not present
	0	CME_ERR_PhonFail	phone failure
	1	CME_ERR_NoConnect	no connection to phone
	2	CME_ERR_LinkRes	phone-adaptor link reserved
	3	CME_ERR_OpNotAllow	operation not allowed
	4	CME_ERR_OpNotSupp	operation not supported
	5	CME_ERR_PhSimPinReq	PH-SIM PIN required
	10	CME_ERR_SimNotIns	SIM not inserted
	11	CME_ERR_SimPinReq	SIM PIN required
	12	CME_ERR_SimPukReq	SIM PUK required
	13	CME_ERR_SimFail	SIM failure
	14	CME_ERR_SimBusy	SIM busy
	15	CME_ERR_SimWrong	SIM wrong
	16	CME_ERR_WrongPasswd	wrong password

enum type	value	symbolic constant	comment
	17	CME_ERR_SimPin2Req	SIM PIN2 required
	18	CME_ERR_SimPuk2Req	SIM PUK2 required
	20	CME_ERR_MemFull	memory full
	21	CME_ERR_InvIdx	invalid index
	22	CME_ERR_NotFound	not found
	23	CME_ERR_MemFail	memory failure
	24	CME_ERR_TxtToLong	text string too long
	25	CME_ERR_InvalidTxtChar	invalid characters in text string
	26	CME_ERR_DialToLong	dial string too long
	27	CME_ERR_InvDialChar	invalid characters in dial string
	30	CME_ERR_NoServ	no network service
	31	CME_ERR_Timeout	network timeout
	32	CME_ERR_LimServ	limited service
	100	CME_ERR_Unknown	unknown
	512	CME_ERR_FailedToAbort	failed to abort
	513	CME_ERR_AcmResetNeeded	ACM reset needed

4.3.36 T_ACI_CMGF_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CMGF_MOD	-1	CMGF_MOD_NotPresent	parameter not present
	0	CMGF_MOD_Pdu	PDU mode (default when implemented)
	1	CMGF_MOD_Txt	text mode

4.3.37 T_ACI_CMOD_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CMOD_MOD	-1	CMOD_MOD_NotPresent	parameter not present
	0	CMOD_MOD_Single	single mode
	1	CMOD_MOD_VoiceFax	alternating voice/fax (teleservice 61)
	2	CMOD_MOD_VoiceDat	alternating voice/data (bearer service 61)
	3	CMOD_MOD_VoiceFlwdDat	voice followed by data (bearer service 81)

4.3.38 T_ACI_CMS_ERR

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CMS_ERR	-1	CMS_ERR_NotPresent	parameter not present
	1	CMS_ERR_UnAllocNum	unassigned (unallocated) number
	8	CMS_ERR_OpDetermBarr	operator determined barring
	10	CMS_ERR_CallBarr	call barred

enum type	value	symbolic constant	comment
	21	CMS_ERR_TransReject	short message transfer rejected
	27	CMS_ERR_DestOutOfServ	destination out of service
	28	CMS_ERR_UnidentSubsc	unidentified subscriber
	29	CMS_ERR_FacReject	facility rejected
	30	CMS_ERR_UnKnownSubsc	unknown subscriber
	38	CMS_ERR_NetOutOfOrder	network out of order
	41	CMS_ERR_TempFail	temporary failure
	42	CMS_ERR_Congestion	congestion
	47	CMS_ERR_ResUnAvail	resources unavailable, unspecified
	50	CMS_ERR_FacNotSubscr	requested facility not subscribed
	69	CMS_ERR_FacNotImpl	requested facility not implemented
	81	CMS_ERR_TransRefInval	invalid short message transfer reference value
	95	CMS_ERR_InValSM	invalid message, unspecified
	96	CMS_ERR_InValManInfo	invalid mandatory information
	97	CMS_ERR_MsgTypNotExist	message type non-existent or not implemented
	98	CMS_ERR_MsgNotCompatible	message not compatible with short message protocol state
	99	CMS_ERR_InfoElemNotImpl	information element non-existent or not implemented
	111	CMS_ERR_ProtErr	protocol error, unspecified
	127	CMS_ERR_InterWrkUnSpec	Inter-working, unspecified
	128	CMS_ERR_TlmtkNotSup	telematic inter-working not supported
	129	CMS_ERR_SMOtNotSup	short message Type 0 not supported
	130	CMS_ERR_CantReplceSM	cannot replace short message
	143	CMS_ERR_UnSpecPIDErr	unspecified TP-PID error
	144	CMS_ERR_DcsNotSup	data coding scheme (alphabet) not supported
	145	CMS_ERR_MsgClassNotSup	message class not supported
	159	CMS_ERR_UnSpecTpDcs	unspecified TP-DCS error
	160	CMS_ERR_CmdNotAct	command cannot be actioned
	161	CMS_ERR_CmdUnSup	command unsupported
	175	CMS_ERR_UnSpecTpCmd	unspecified TP Command error
	176	CMS_ERR_TpduUnSup	TPDU not supported
	192	CMS_ERR_ScBsy	SC busy
	193	CMS_ERR_NoScSubsc	no SC subscription
	194	CMS_ERR_ScSysFail	SC system failure
	195	CMS_ERR_InValSme	invalid SME address
	196	CMS_ERR_DestSmeBarr	destination SME barred
	197	CMS_ERR_SmRejectDuplSm	SM Rejected-Duplicate SM

enum type	value	symbolic constant	comment
	208	CMS_ERR_SimSmsStorFull	SIM SMS storage full
	209	CMS_ERR_NoStorInSim	no SMS storage capability in SIM
	210	CMS_ERR_ErrInMs	error in MS
	211	CMS_ERR_MemCabExcee	memory capacity exceeded
	255	CMS_ERR_UnSpecErr	unspecified error cause
	300	CMS_ERR_MeFail	ME failure
	301	CMS_ERR_ServRes	SMS service of ME reserved
	302	CMS_ERR_OpNotAllowed	operation not allowed
	303	CMS_ERR_OpNotSup	operation not supported
	304	CMS_ERR_InValPduMod	invalid PDU mode parameter
	305	CMS_ERR_InValTxtMod	invalid text mode parameter
	310	CMS_ERR_SimNotIns	SIM not inserted
	311	CMS_ERR_SimPinReq	SIM PIN necessary
	312	CMS_ERR_PhSimPinReq	PH-SIM PIN necessary
	313	CMS_ERR_SimFail	SIM failure
	314	CMS_ERR_SimBsy	SIM busy
	315	CMS_ERR_SimWrong	SIM wrong
	316	CMS_ERR_SIMPukReq	SIM PUK required
	317	CMS_ERR_SIMPin2Req	SIM PIN2 required
	318	CMS_ERR_SIMPuk2Req	SIM PUK2 required
	320	CMS_ERR_MemFail	memory failure
	321	CMS_ERR_InValMemIdx	invalid memory index
	322	CMS_ERR_MemFull	memory full
	330	CMS_ERR_SmscAdrUnKnown	SMSC address unknown
	331	CMS_ERR_NoNetServ	no network service
	332	CMS_ERR_NetTimeOut	network timeout
	340	CMS_ERR_NoCnmaAckExpect	no CNMA ack expected
	500	CMS_ERR_UnknownErr	unknown error
	512	CMS_ERR_FailedToAbort	failed to abort
	513	CMS_ERR_AcmResetNeeded	ACM reset needed

4.3.39 T_ACI_CNMI_MT

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CNMI_MT	-1	CNMI_MT_NotPresent	parameter not present
	0	CNMI_MT_NoSmsDeliverInd	store message according to its class, do not generate SMS-DELIVER indication
	1	CNMI_MT_SmsDeliverInd	store message according to its class, generate SMS-DELIVER indication
	2	CNMI_MT_SmsDeliver	forward message to TE (except class 2: like

enum type	value	symbolic constant	comment
	3	CNMI_MT_SmsDeliverCls3	<mt>=1) forward message to TE (except class 0,1,2: like <mt>=1)

4.3.40 T_ACI_CNMI_BM

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CNMI_BM	-1	CNMI_BM_NotPresent	parameter not present
	0	CNMI_BM_NoCbmInd	store message to BM, do not generate CBM indication
	1	CNMI_BM_CbmInd	store message to BM, generate CBM indication
	2	CNMI_BM_Cbm	forward message to TE
	3	CNMI_BM_CbmCls3	forward message to TE (except class 0,1,2: like <bm>=1)

4.3.41 T_ACI_CNMI_DS

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CNMI_DS	-1	CNMI_DS_NotPresent	parameter not present
	0	CNMI_DS_NoSmsStatRpt	Do not forward SMS-STATUS-REPORT to TE
	1	CNMI_DS_SmsStatRpt	Forward SMS-STATUS-REPORT to TE <bfr>

4.3.42 T_ACI_CNUM_ITC

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CNUM_ITC	-1	CNUM_ITC_NotPresent	parameter not present
	0	CNUM_ITC_3_1_kHz	3.1 kHz
	1	CNUM_ITC_Udi	UDI

4.3.43 T_ACI_CNUM_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CNUM_MOD	0	CNUM_MOD_NewRead	read first entry
	1	CNUM_MOD_NextRead	read next entry

4.3.44 T_ACI_CNUM_SERV

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CNUM_SERV	-1	CNUM_SERV_NotPresent	parameter not present
	0	CNUM_SERV_Asynch	asynchronous modem
	1	CNUM_SERV_Synch	synchronous modem

enum type	value	symbolic constant	comment
	2	CNUM_SERV_PadAsynch	PAD access (asynchronous)
	3	CNUM_SERV_PacketSynch	packet access (synchronous)
	4	CNUM_SERV_Voice	voice
	5	CNUM_SERV_Fax	fax

4.3.45 T_ACI_COLP_STAT**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_COLP_STAT	-1	COLP_STAT_NotPresent	parameter not present
	0	COLP_STAT_NotProv	COLP not provisioned
	1	COLP_STAT_Prov	COLP provisioned
	2	COLP_STAT_Unknown	unknown (e.g. no network, etc.)

4.3.46 T_ACI_COLR_STAT**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_COLR_STAT	-1	COLR_STAT_NotPresent	parameter not present
	0	COLR_STAT_NotProv	not provisioned
	1	COLR_STAT_Prov	provisioned
	2	COLR_STAT_Unknown	unknown

4.3.47 T_ACI_COPN_LID**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_COPN_LID	-1	COPN_LID_NotPresent	parameter not present
	0	COPN_LID_Pcm	list in permanent configuration mem
	1	COPN_LID_Cnst	list in constant memory

4.3.48 T_ACI_COPS_FRMT**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_COPS_FRMT	-1	COPS_FRMT_NotPresent	parameter not present
	0	COPS_FRMT_Long	long format alphanumeric
	1	COPS_FRMT_Short	short format alphanumeric
	2	COPS_FRMT_Numeric	numeric

4.3.49 T_ACI_COPS_MOD**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_COPS_MOD	-1	COPS_MOD_NotPresent	parameter not present

enum type	value	symbolic constant	comment
	0	COPS_MOD_Auto	automatic
	1	COPS_MOD_Man	manual
	2	COPS_MOD_Dereg	deregister from network
	3	COPS_MOD_SetOnly	set operator format only
	4	COPS_MOD_Both	manual/automatic, if manual selection fails, automatic mode is entered

4.3.50 T_ACI_COPS_STAT**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_COPS_STAT	-1	COPS_STAT_NotPresent	parameter not present
	0	COPS_STAT_Unknown	unknown
	1	COPS_STAT_Available	available
	2	COPS_STAT_Current	current
	3	COPS_STAT_Forbidden	forbidden

4.3.51 T_ACI_CPAS_PAS**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CPAS_PAS	-1	CPAS_PAS_NotPresent	parameter not present
	0	CPAS_PAS_Ready	ready
	1	CPAS_PAS_Unavailable	unavailable
	2	CPAS_PAS_Unknown	unknown
	3	CPAS_PAS_Ring	ringing
	4	CPAS_PAS_CallProg	call in progress
	5	CPAS_PAS_Asleep	asleep

4.3.52 T_ACI_CPBF_MOD**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CPBF_MOD	0	CPBF_MOD_NewSearch	start a new search
	1	CPBF_MOD_NextSearch	continue a started search

4.3.53 T_ACI_CPI_IBT**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CPI_IBT	-1	CPI_IBT_NotPresent	parameter not present
	0	CPI_IBT_False	use of in-band tones false
	1	CPI_IBT_True	use of in-band tones true

4.3.54 T_ACI_CPI_MSG**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CPI_MSG	-1	CPI_MSG_NotPresent	parameter not present
	0	CPI_MSG_Setup	SETUP message
	1	CPI_MSG_Disc	DISCONNECT message
	2	CPI_MSG_Alert	ALERTING message
	3	CPI_MSG_Proc	CALL PROCEEDING message
	4	CPI_MSG_Sync	SYNCRONIZATION message

4.3.55 T_ACI_CPI_TCH**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CPI_TCH	-1	CPI_TCH_NotPresent	parameter not present
	0	CPI_TCH_False	traffic channel assigned false
	1	CPI_TCH_True	traffic channel assigned true

4.3.56 T_ACI_CPIN_RSLT**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CPIN_RSLT	-1	CPIN_RSLT_NotPresent	parameter not present
	0	CPIN_RSLT_SimReady	ME is not pending for a password
	1	CPIN_RSLT_SimPinReq	ME is waiting SIM PIN to be provided
	2	CPIN_RSLT_SimPukReq	ME is waiting SIM PUK to be provided
	4	CPIN_RSLT_SimPin2Req	ME is waiting SIM PIN2 to be provided
	5	CPIN_RSLT_SimPuk2Req	ME is waiting SIM PUK2 to be provided

4.3.57 T_ACI_CPOL_FRMT**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CPOL_FRMT	-1	CPOL_FRMT_NotPresent	parameter not present
	0	CPOL_FRMT_Long	long format alphanumeric
	1	CPOL_FRMT_Short	short format alphanumeric
	2	CPOL_FRMT_Numeric	numeric

4.3.58 T_ACI_CPWD_FAC**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_CPWD_FAC	-1	CPWD_FAC_NotPresent	parameter not present
	2	CPWD_FAC_Sc	SIM (lock SIM card) (SIM asks for password in ME power-up and when this lock command issued)

enum type	value	symbolic constant	comment
	3	CPWD_FAC_Ao	BAOC (Barr All Outgoing Calls)
	4	CPWD_FAC_Oi	BOIC (Barr Outgoing International Calls)
	5	CPWD_FAC_Ox	BOIC-exHC (Barr Outgoing International Calls except to Home Country)
	6	CPWD_FAC_Ai	BAIC (Barr All Incoming Calls)
	7	CPWD_FAC_Ir	BIC-Roam (Barr Incoming Calls when roaming outside the home country)
	12	CPWD_FAC_Ab	All Barring services
	13	CPWD_FAC_Ag	All outGoing barring services
	14	CPWD_FAC_Ac	All inComing barring services
	15	CPWD_FAC_Fd	SIM fixed dialing memory feature (if PIN2 authentication has not been done during the current session, PIN2 is required as <passwd>)
	16	CPWD_FAC_P2	SIM PIN 2

4.3.59 T_ACI_CR_SERV

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CR_SERV	-1	CR_SERV_NotPresent	parameter not present
	0	CR_SERV_Async	asynchronous transparent
	1	CR_SERV_Sync	asynchronous transparent
	2	CR_SERV_RelAsync	asynchronous non-transparent
	3	CR_SERV_RelSync	synchronous non-transparent

4.3.60 T_ACI_CREG_STAT

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CREG_STAT	-1	CREG_STAT_NotPresent	parameter not present
	0	CREG_STAT_NoSearch	not registered, ME is not currently searching for a new operator with which to register
	1	CREG_STAT_Reg	registered, home network
	2	CREG_STAT_Search	not registered, but ME is currently searching for a new operator with which to register
	3	CREG_STAT_Denied	registration denied
	4	CREG_STAT_Unknown	unknown
	5	CREG_STAT_Roam	registered, roaming

4.3.61 T_ACI_CRING_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CRING_MOD	-1	CRING_MOD_NotPresent	parameter not present

enum type	value	symbolic constant	comment
	0	CRING_MOD_Direct	direct
	1	CRING_MOD_Alternate	alternating

4.3.62 T_ACI_CRING_TYP

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CRING_TYP	-1	CRING_TYP_NotPresent	parameter not present
	0	CRING_TYP_Async	asynchronous transparent
	1	CRING_TYP_Sync	asynchronous transparent
	2	CRING_TYP_RelAsync	asynchronous non-transparent
	3	CRING_TYP_RelSync	synchronous non-transparent
	4	CRING_TYP_Fax	facsimile
	5	CRING_TYP_Voice	voice

4.3.63 T_ACI_CRSM_CMD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CRSM_CMD	-1	CRSM_CMD_NotPresent	parameter not present
	176	CRSM_CMD_ReadBin	read binary
	178	CRSM_CMD_ReadRec	read record
	192	CRSM_CMD_GetResp	get response
	214	CRSM_CMD_UpdBin	update binary
	220	CRSM_CMD_UpdRec	update record
	242	CRSM_CMD_Status	get status

4.3.64 T_ACI_CSCB_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CSCB_MOD	-1	CSCB_MOD_NotPresent	parameter not present
	0	CSCB_MOD_Accept	message types specified in <mids> and <dcss> are accepted
	1	CSCB_MOD_NotAccept	message types specified in <mids> and <dcss> are not accepted

4.3.65 T_ACI_CSCS_CHSET

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CSCS_CHSET	-1	CSCS_CHSET_NotPresent	parameter not present
	0	CSCS_CHSET_Ira	international reference alphabet (ITU-T T.50 [13])
	1	CSCS_CHSET_Pcdn	PC Danish/Norwegian character set

enum type	value	symbolic constant	comment
	2	CSCS_CHSET_8859_1	ISO 8859 Latin 1 character set
	3	CSCS_CHSET_Pcp_437	PC character set Code Page 437
	4	CSCS_CHSET_Gsm	GSM default alphabet (GSM 03.38)
	5	CSCS_CHSET_Hex	character strings consist of only hexadecimal numbers from 00 to FF

4.3.66 T_ACI_CSMS_SERV

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CSMS_SERV	-1	CSMS_SERV_NotPresent	parameter not present
	0	CSMS_SERV_GsmPh2	GSM 03.40 and 03.41 (the syntax of SMS AT Commands is compatible with GSM 07.05 Phase 2 version 4.7.0)
	1	CSMS_SERV_GsmPh2Plus	GSM 03.40 and 03.41 (the syntax of SMS AT Commands is compatible with GSM 07.05 Phase 2+ version)

4.3.67 T_ACI_CSMS_SUPP

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CSMS_SUPP	-1	CSMS_SUPP_NotPresent	parameter not present
	0	CSMS_SUPP_Disable	type not supported
	1	CSMS_SUPP_Enable	type supported

4.3.68 T_ACI_CSNS_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CSNS_MOD	-1	CSNS_MOD_NotPresent	parameter not present
	0	CSNS_MOD_Voice	voice
	1	CSNS_MOD_VAFVoice	alternating voice/fax, voice first
	2	CSNS_MOD_Fax	fax
	3	CSNS_MOD_VADVoice	alternating voice/data, voice first
	4	CSNS_MOD_Data	data
	5	CSNS_MOD_VAFFax	alternating voice/fax, fax first
	6	CSNS_MOD_VADData	alternating voice/data, data first
	7	CSNS_MOD_VFD	voice followed by data

4.3.69 T_ACI_CSSI_CODE

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CSSI_CODE	-1	CSSI_CODE_NotPresent	parameter not present
	0	CSSI_CODE_CFUActive	unconditional call forw. actice

enum type	value	symbolic constant	comment
	1	CSSI_CODE_SomeCCFActive	some conditional call forw. active
	2	CSSI_CODE_ForwardedCall	call has been forwarded
	3	CSSI_CODE_CallWaiting	call waiting
	4	CSSI_CODE_CUGCall	closed user group call
	5	CSSI_CODE_OutCallsBarred	outgoing calls barred
	6	CSSI_CODE_IncCallsBarred	incoming calls barred
	7	CSSI_CODE_CLIRSupRej	CLIR suppression rejected
	8	CSSI_CODE_DeflectedCall	call has been deflected

4.3.70 T_ACI_CSSU_CODE

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CSSU_CODE	-1	CSSU_CODE_NotPresent	parameter not present
	0	CSSU_CODE_ForwardedCall	call forwarded
	1	CSSU_CODE_CUGCall	closed user group call
	2	CSSU_CODE_OnHold	call has been put on hold
	3	CSSU_CODE_Retrieved	call has been retrieved
	4	CSSU_CODE_Multiparty	multiparty call entered
	5	CSSU_CODE_HeldCallRel	call on hold has been released
	6	CSSU_CODE_FwrddCheckSS	forward check SS message received
	7	CSSU_CODE_ECTAlert	call is being connected (alerting) with the remote party in alerting state in explicit call transfer operation (during a voice call)
	8	CSSU_CODE_ECTConnect	call has been connected with the other remote party in explicit call transfer operation (also number and subaddress parameters may be present) (during a voice call or MT call setup)
	9	CSSU_CODE_DeflectedCall	deflected call (MT call setup)

4.3.71 T_ACI_CUSD_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_CUSD_MOD	-1	CUSD_MOD_NotPresent	parameter not present
	0	CUSD_MOD_NoActReq	no further user action required
	1	CUSD_MOD_YesActReq	further user action required

4.3.72 T_ACI_D_CLIR_OVRD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_D_CLIR_OVRD	-1	D_CLIR_OVRD_Default	the default setting for CLIR will be used for the next call

enum type	value	symbolic constant	comment
	0	D_CLIR_OVRD_Supp	CLIR will be suppressed for the next call
	1	D_CLIR_OVRD_Invoc	CLIR will be invoked for the next call

4.3.73 T_ACI_D_CUG_CTRL

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_D_CUG_CTRL	-1	D_CUG_CTRL_NotPresent	parameter not present
	0	D_CUG_CTRL_Present	parameter present

4.3.74 T_ACI_D_TOC

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_D_TOC	-1	D_TOC_Data	type of call data
	0	D_TOC_Voice	type of call voice

4.3.75 T_ACI_DR_TYP

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_DR_TYP	-1	DR_TYP_NotPresent	parameter not present
	0	DR_TYP_None	data compression is not in use
	1	DR_TYP_TxOnly	data compression is in use in transmit direction only
	2	DR_TYP_RxOnly	data compression is in use in receive direction only
	3	DR_TYP_Both	data compression is in use in both directions

4.3.76 T_ACI_DS_COMP

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_DS_COMP	-1	DS_COMP_NotPresent	parameter not present
	0	DS_COMP_DoNotDisc	do not disconnect if Rec. V.42 <i>bis</i> is not negotiated by the remote DCE
	1	DS_COMP_Disc	disconnect if Rec. V.42 <i>bis</i> is not negotiated by the remote DCE

4.3.77 T_ACI_DS_DIR

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_DS_DIR	-1	DS_DIR_NotPresent	parameter not present
	0	DS_DIR_Negotiated	negotiated ... no compression
	1	DS_DIR_TxOnly	transmit only

enum type	value	symbolic constant	comment
	2	DS_DIR_RxOnly	receive only
	3	DS_DIR_Both	both directions

4.3.78 T_ACI_ERR_CLASS

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_ERR_CLASS	-1	ACI_ERR_CLASS_NotPresent	parameter not present
	0	ACI_ERR_CLASS_Cme	classifies +CME error codes
	1	ACI_ERR_CLASS_Cms	classifies +CMS error codes
	2	ACI_ERR_CLASS_Ceer	classifies +CEER error codes
	3	ACI_ERR_CLASS_Ext	classifies extended error codes

4.3.79 T_ACI_EXT_ERR

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_EXT_ERR	-1	EXT_ERR_NotPresent	parameter not present
	0	EXT_ERR_Parameter	wrong parameter setting
	1	EXT_ERR_DataCorrupt	corrupted data found
	2	EXT_ERR_Internal	internal failure
	3	EXT_ERR_CallTabFull	call table is full
	4	EXT_ERR_SrvTabFull	service table is full
	5	EXT_ERR_CallNotFound	call was not found in call table
	6	EXT_ERR_NoDataCallSup	no data calls are supported
	7	EXT_ERR_OneCallOnHold	one call is already on hold
	8	EXT_ERR_CallTypeNoHold	type of call could not be put on hold
	9	EXT_ERR_FdnCheck	FDN number check failed
	10	EXT_ERR_BdnCheck	BDN number check failed
	11	EXT_ERR_ParallelUSSD	parallel USSD transaction
	12	EXT_ERR_FaxMinSpeedCond	minimum speed condition for FAX
	13	EXT_ERR_CmdDetailsSAT	SAT command details failure

4.3.80 T_ACI_KSD_CMD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_KSD_CMD	200	KSD_CMD_NONE	no KSD command identifier
	201	KSD_CMD_CB	call barring command id
	202	KSD_CMD_CF	call forwarding command id
	203	KSD_CMD_CL	calling line command id
	204	KSD_CMD_CW	call waiting command id
	205	KSD_CMD_PWD	password command id

enum type	value	symbolic constant	comment
	206	KSD_CMD_UBLK	unlock PIN command id
	207	KSD_CMD_USSD	unstructured SS command id
	208	KSD_CMD_IMEI	get IMEI command id
	209	KSD_CMD_MAX	maximum command id

4.3.81 T_ACI_NRG_FRMT

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_NRG_FRMT	-1	NRG_FRMT_NotPresent	parameter not present
	0	NRG_FRMT_Long	long format alphanumeric
	1	NRG_FRMT_Short	short format alphanumeric
	2	NRG_FRMT_Numeric	numeric

4.3.82 T_ACI_NRG_RGMD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_NRG_RGMD	-1	NRG_RGMD_NotPresent	parameter not present
	0	NRG_RGMD_Auto	automatic registration mode
	1	NRG_RGMD_Manual	manual registration mode

4.3.83 T_ACI_NRG_SVMD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_NRG_SVMD	-1	NRG_SVMD_NotPresent	parameter not present
	0	NRG_SVMD_Full	full service
	1	NRG_SVMD_Limited	limited service
	2	NRG_SVMD_NoSrv	no service
	3	NRG_SVMD_SetRegModeOnly	set registration mode only

4.3.84 T_ACI_PB_STOR

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_PB_STOR	-1	PB_STOR_NotPresent	parameter not present
	0	PB_STOR_Fd	SIM fixdialing phone book
	1	PB_STOR_Ld	SIM last-dialing phone book
	2	PB_STOR_Ed	emergency call numbers
	3	PB_STOR_Ad	abbreviated dialing numbers
	4	PB_STOR_Bd	barred dialing numbers
	5	PB_STOR_Lr	last received numbers
	6	PB_STOR_Sd	service dialing numbers

enum type	value	symbolic constant	comment
	7	PB_STOR_Lm	last missed numbers
	8	PB_STOR_Af	combination of fixed and abbreviated dialling phonebook

4.3.85 T_ACI_PBCF_LDN**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_PBCF_LDN	-1	PBCF_LDN_NotPresent	parameter not present
	0	PBCF_LDN_Enable	last dialled number enable
	1	PBCF_LDN_Disable	last dialled number disable

4.3.86 T_ACI_PBCF_LRN**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_PBCF_LRN	-1	PBCF_LRN_NotPresent	parameter not present
	0	PBCF_LRN_Enable	last received number enable
	1	PBCF_LRN_Disable	last received number disable

4.3.87 T_ACI_PVRF_STAT**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_PVRF_STAT	-1	PVRF_STAT_NotPresent	parameter not present
	0	PVRF_STAT_NotRequired	not required
	1	PVRF_STAT_Required	required

4.3.88 T_ACI_PVRF_TYPE**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_PVRF_TYPE	-1	PVRF_TYPE_NotPresent	parameter not present
	0	PVRF_TYPE_Pin1	type PIN1
	1	PVRF_TYPE_Pin2	type PIN2
	2	PVRF_TYPE_Puk1	type PUK1
	3	PVRF_TYPE_Puk2	type PUK2

4.3.89 T_ACI_RETURN**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_RETURN	-1	AT_FAIL	failed
	0	AT_CMPL	successfully completed
	1	AT_EXCT	executing
	2	AT_BUSY	command handler is busy

4.3.90 T_ACI_RLOG_RSLT**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_RLOG_RSLT	-1	RLOG_RSLT_NotPresent	parameter not present
	0	RLOG_RSLT_OK	result was "OK"
	1	RLOG_RSLT_NoCarrier	result was "NO CARRIER"
	2	RLOG_RSLT_Connect	result was "CONNECT"
	3	RLOG_RSLT_Busy	result was "BUSY"
	4	RLOG_RSLT_NoAnswer	result was "NO ANSWER"
	5	RLOG_RSLT_CME	result was "ERROR"

4.3.91 T_ACI_SATT_CS**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_SATT_CS	-1	SATT_CS_NotPresent	parameter not present
	0	SATT_CS_UserRedialStop	user redial stop
	1	SATT_CS_EndRedial	end of redial
	2	SATT_CS_EndSession	end of session

4.3.92 T_ACI_SMS_READ**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_SMS_READ	-1	SMS_READ_NotPresent	parameter not present
	0	SMS_READ_Normal	normal read with status change
	1	SMS_READ_Preview	read without status change
	2	SMS_READ_StatusChange	change status to read

4.3.93 T_ACI_SMS_STAT**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_SMS_STAT	-1	SMS_STAT_NotPresent	parameter not present
	0	SMS_STAT_RecUnread	received unread message
	1	SMS_STAT_RecRead	received read message
	2	SMS_STAT_StoUnsent	stored unsent message
	3	SMS_STAT_StoSent	stored sent message
	4	SMS_STAT_All	all messages

4.3.94 T_ACI_SMS_STOR**Type Declaration:**

enum type	value	symbolic constant	comment
T_ACI_SMS_STOR	-1	SMS_STOR_NotPresent	parameter not present

enum type	value	symbolic constant	comment
	0	SMS_STOR_Me	ME message storage
	1	SMS_STOR_Sm	SIM message storage

4.3.95 T_ACI_STATUS

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_STATUS	-1	STATUS_NotPresent	parameter not present
	0	STATUS_NotActive	not active
	1	STATUS_Active	active

4.3.96 T_ACI_TOA_NPI

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_TOA_NPI	-1	NPI_NotPresent	parameter not present
	0	NPI_Unknown	unknown
	1	NPI_IsdnTelephony	ISDN/telephony numbering plan (Rec. E.164/E.163)
	3	NPI_Data	data numbering plan (Rec. X.121)
	4	NPI_Telex	telex numbering plan (Rec. F.69)
	8	NPI_National	national numbering plan
	9	NPI_Private	private numbering plan

4.3.97 T_ACI_TOA_TON

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_TOA_TON	-1	TON_NotPresent	parameter not present
	0	TON_Unknown	unknown
	1	TON_International	international number
	2	TON_National	national number
	3	TON_NetSpecific	network specific number
	4	TON_DedAccess	dedicated access, short code

4.3.98 T_ACI_TOS_OE

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_TOS_OE	-1	OE_NotPresent	parameter not present
	0	OE_Even	even
	1	OE_Odd	odd

4.3.99 T_ACI_TOS_TOS

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_TOS_TOS	-1	TOS_NotPresent	parameter not present
	0	TOS_Nsap	NSAP (X.213/ISO 8348 AD2)
	2	TOS_User	User-specified

4.3.100 T_ACI_VTS_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_VTS_MOD	-1	VTS_MOD_NotPresent	parameter not present
	0	VTS_MOD_Auto	send DTMF with fixed length
	1	VTS_MOD_ManStart	manual start sending DTMF
	2	VTS_MOD_ManStop	manual stop sending DTMF

4.3.101 T_ACI_WS46_MOD

Type Declaration:

enum type	value	symbolic constant	comment
T_ACI_WS46_MOD	-1	WS46_MOD_NotPresent	parameter not present
	12	WS46_MOD_Gsm	GSM digital cellular

4.4 Structures

4.4.1 T_ACI_CAL_ENTR

Type Declaration:

struct type	element	element type	comment
T_ACI_CAL_ENTR	index	SHORT	index in call table
	status	T_ACI_CAL_STAT	status of call
	number	CHAR[MAX_NUM_LEN]	phone number of format <type>
	type	T_ACI_TOA	type of address
	alpha	CHAR[MAX_ALPHA_LEN]	alphanumeric phone book entry corresponding to phone number
	calType	T_ACI_CAL_TYPE	type of call
	ibtUse	T_ACI_CAL_IBT	use of in-band tones
	calMode	T_ACI_CAL_MODE	mode of call
	calOwner	T_ACI_CAL_OWN	owner of call
	mpty	T_ACI_CAL_MPTY	multiparty status

4.4.2 T_ACI_CBM_DATA

Type Declaration:

struct type	element	element type	comment
T_ACI_CBM_DATA	data	UBYTE[MAX_CBM_LEN]	cell broadcast message data
	len	UBYTE	length of data

4.4.3 T_ACI_CCFC_SET

Type Declaration:

struct type	element	element type	comment
T_ACI_CCFC_SET	clsstat	T_ACI_CLSSTAT	...
	number	CHAR[MAX_NUM_LEN]	string type phone number of forwarding address in format specified by <type>
	type	T_ACI_TOA	type of address
	subaddr	CHAR[MAX_SUBADDR_LEN]	subaddress of format specified by <satype>
	satype	T_ACI_TOS	type of subaddress
	time	SHORT	when no reply is enabled or queried, this gives the time in seconds to wait before call is forwarded

4.4.4 T_ACI_CLCC_CALDESC

Type Declaration:

struct type	element	element type	comment
T_ACI_CLCC_CALDESC	idx	SHORT	call identification number

struct type	element	element type	comment
	dir	T_ACI_CLCC_DIR	direction (MO/MT)
	stat	T_ACI_CLCC_STAT	status of call
	mode	T_ACI_CLCC_MODE	mode (bearer/teleservice)
	mpty	T_ACI_CLCC_MPTY	multiparty, indicates whether call is conference call
	number	CHAR[MAX_NUM_LEN]	phone number in format specified by type
	type	T_ACI_TOA	type of address
	alpha	CHAR[MAX_ALPHA_LEN]	number in alphanumeric format corresponding to the entry found in phonebook

4.4.5 T_ACI_CLOG

Type Declaration:

struct type	element	element type	comment
T_ACI_CLOG	atCmd	T_ACI_AT_CMD	AT command, doesn't distinguish between set, query and test
	cmdType	T_ACI_CLOG_TYPE	type of AT command (set, query, test)
	retCode	T_ACI_RETURN	return code of AT command
	cId	SHORT	call identifier
	sID	SHORT	service identifier
	cmdPrm	union	command parameter

4.4.5.1 cmdPrm

Definition:

union type	element	element type	comment
Beware! The elements here share the same memory location because this is a union.	sCFUN	struct	used if sAT_PlusCFUN ()
	sCPIN	struct	used if sAT_PlusCPIN ()
	sCOPS	struct	used if sAT_PlusCOPS ()
	tCOPS	struct	used if tAT_PlusCOPS ()
	qCLIP	struct	used if qAT_PlusCLIP ()
	qCLIR	struct	used if qAT_PlusCLIR ()
	qCOLP	struct	used if qAT_PlusCOLP ()
	sD	struct	used if sAT_Dm () or sAT_Dn ()
	sA	struct	used if sAT_A ()
	sH	struct	used if sAT_H ()
	sCHUP	struct	used if sAT_PlusCHUP ()
	sCLCK	struct	used if sAT_PlusCLCK ()
	qCLCK	struct	used if qAT_PlusCLCK ()
	sCPWD	struct	used if sAT_PlusCPWD ()
	sCCFC	struct	used if sAT_PlusCCFC ()

union type	element	element type	comment
	qCCFC	struct	used if qAT_PlusCCFC ()
	sCTRF	struct	not implemented yet
	sCCWA	struct	used if sAT_PlusCCWA ()
	qCCWA	struct	used if qAT_PlusCCWA ()
	sCHLD	struct	used if sAT_PlusCHLD ()
	sCUSD	struct	used if sAT_PlusCUSD ()
	sNRG	struct	used if sAT_PercentNRG ()
	qCOLR	struct	used if sAT_PercentCOLR ()
	sCMGD	struct	used if sAT_PlusCMGD ()
	sCMGW	struct	used if sAT_PlusCMGW ()
	sCSMS	struct	used if sAT_PlusCSMS ()
	sCPMS	struct	used if sAT_PlusCPMS ()
	sCMGF	struct	used if sAT_PlusCMGF ()
	qCMGF	struct	used if qAT_PlusCMGF ()
	sCSCB	struct	used if sAT_PlusCSCB ()
	sCMGS	struct	used if sAT_PlusCMGS ()
	sCMSS	struct	used if sAT_PlusCMSS ()
	sCMGC	struct	used if sAT_PlusCMGC ()
	sKSCF	struct	used with sAT_D...
	sKSCB	struct	used with sAT_D...
	sKSCW	struct	used with sAT_D...
	sKSCL	struct	used with sAT_D...
	sKSPW	struct	used with sAT_D...
	sKSUB	struct	used with sAT_D...
	sKSUS	struct	used with sAT_D...
	sPVRF	struct	used if sAT_PercentPVRF ()

4.4.5.1.1 sCFUN

Definition:

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	fun	T_ACI_CFUN_FUN	phone functionality
	rst	T_ACI_CFUN_RST	reset parameter

4.4.5.1.2 sCPIN

Definition:

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	pin	CHAR*	PIN
	newpin	CHAR*	newpin

4.4.5.1.3 sCOPS**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	mode	T_ACI_COPS_MOD	registration mode
	format	T_ACI_COPS_FRMT	format of network operator
	oper	CHAR*	network operator

4.4.5.1.4 tCOPS**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	startIdx	SHORT	start index
	lastIdx	SHORT*	last index
	operLst	T_ACI_COPS_OPDESC*	network operator list

4.4.5.1.5 qCLIP**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	stat	T_ACI_CLIP_STAT*	status

4.4.5.1.6 qCLIR**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	mode	T_ACI_CLIR_MOD*	mode
	stat	T_ACI_CLIR_STAT*	status

4.4.5.1.7 qCOLP**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	stat	T_ACI_COLP_STAT*	status

4.4.5.1.8 sD**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	number	CHAR*	phone number
	T_ACI_D_CLIR_OVRD	clirOvrd	calling line identification restriction override

struct type	element	element type	comment
	T_ACI_D_CUG_CTRL	cugCtrl	closed user group control
	T_ACI_D_TOC	callType	type of call (voice/no voice)

4.4.5.1.9 sA**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source

4.4.5.1.10 sH**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source

4.4.5.1.11 sCHUP**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source

4.4.5.1.12 sCLCK**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	fac	T_ACI_CLCK_FAC	ME or network facility
	mode	T_ACI_CLCK_MOD	locking mode
	passwd	CHAR*	ME or network facility password
	class	T_ACI_CLASS	service class

4.4.5.1.13 qCLCK**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	fac	T_ACI_CLCK_FAC	ME or network facility
	class	T_ACI_CLASS	service class

4.4.5.1.14 sCPWD**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	fac	T_ACI_CPWD_FAC	ME or network facility
	oldpwd	CHAR*	old password for ME or network facility

struct type	element	element type	comment
	newpwd	CHAR*	new password for ME or network facility

4.4.5.1.15 sCCFC**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	reason	T_ACI_CCFC_RSN	forwarding reason
	mode	T_ACI_CCFC_MOD	forwarding mode
	number	CHAR*	forwarding address
	type	T_ACI_TOA*	type of forwarding address
	class	T_ACI_CLASS	class of information
	subaddr	CHAR*	subaddress
	satype	T_ACI_TOS*	type of subaddress
	SHORT	time	wait time before forwarding

4.4.5.1.16 qCCFC**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	reason	T_ACI_CCFC_RSN	requested call forwarding condition
	class	T_ACI_CLASS	service class

4.4.5.1.17 sCTRF – not supported yet**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	number	CHAR*	phone number
	type	T_ACI_TOA*	type of address
	subaddr	CHAR*	subaddress
	satype	T_ACI_TOS*	type of subaddress

4.4.5.1.18 sCCWA**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	mode	T_ACI_CCWA_MOD	call waiting mode
	class	T_ACI_CLASS	class of information

4.4.5.1.19 qCCWA**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	class	T_ACI_CLASS	service class

4.4.5.1.20 sCHLD**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	mode	T_ACI_CHLD_MOD	action identifier
	call	CHAR*	call index or phone number

4.4.5.1.21 sCUSD**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	str	T_ACI_USSD_DATA*	USSD string
	dcs	SHORT	data coding scheme

4.4.5.1.22 sNRG**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	regMode	T_ACI_NRG_RGMD	registration mode
	srvMode	T_ACI_NRG_SVMD	service mode
	oprFrmt	T_ACI_NRG_FRMT	format for network operator
	opr	CHAR*	network operator

4.4.5.1.23 qCOLR**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	stat	T_ACI_COLR_STAT*	connected line restriction state

4.4.5.1.24 sCMGD**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	index	UBYTE	message storage location

4.4.5.1.25 sCMGW**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	index	SHORT	index of location area to be written
	address	CHAR*	destination/originating address
	toa	T_ACI_TOA*	type of destination/originating address
	stat	T_ACI_SMS_STAT	message state
	data	CHAR*	message data

4.4.5.1.26 sCSMS**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	service	T_ACI_CSMS_SERV	message service

4.4.5.1.27 sCPMS**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	mem1	T_ACI_SMS_STOR	memory from which messages are read and deleted
	mem2	T_ACI_SMS_STOR	memory to which writing and sending operations are made
	mem3	T_ACI_SMS_STOR	memory to which received SMs are preferred to be stored

4.4.5.1.28 sCMGF**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	mode	T_ACI_CMGF_MODE	input and output format of message

4.4.5.1.29 qCMGF**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	mode	T_ACI_CMGF_MODE*	mode of message format

4.4.5.1.30 sCSCB**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	mode	T_ACI_CSCB_MODE	mode of acceptance

struct type	element	element type	comment
	mids	USHORT*	possible combinations of CBM message identifiers
	dcss	UBYTE*	possible combinations of CBM data coding schemes

4.4.5.1.31 sCMGS**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	da	CHAR*	destination address
	toda	T_ACI_TOA*	type of destination address
	data	CHAR*	message data

4.4.5.1.32 sCMSS**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	index	UBYTE	message storage location
	da	CHAR*	destination address
	toda	T_ACI_TOA*	type of destination address

4.4.5.1.33 sCMGC**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	fo	SHORT	first octet of SMS-DELIVER, SMS-SUBMIT, SMS-STATUS-REPORT or SMS-COMMAND
	ct	SHORT	TP-Command-Type
	pid	SHORT	TP-Protocol-Identifier
	mn	SHORT	TP-Message-Number
	da	CHAR*	destination address
	toda	T_ACI_TOA*	type of destination address
	data	CHAR*	message data

4.4.5.1.34 sKSCF**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	opCd	UBYTE	operation code
	ssCd	UBYTE	supplementary service code

struct type	element	element type	comment
	bsTp	UBYTE	basic service type
	bsCd	UBYTE	basic service code
	num	UBYTE*	phone number
	npi	UBYTE	numbering plan
	ton	UBYTE	type of number
	sub	UBYTE*	subaddress
	tos	UBYTE	type of subaddress
	oe	UBYTE	odd/even indicator
	time	UBYTE	no reply time

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.5.1.35 sKSCB

Definition:

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	opCd	UBYTE	operation code
	ssCd	UBYTE	supplementary service code
	bsTp	UBYTE	basic service type
	bsCd	UBYTE	basic service code
	pwd	UBYTE*	password

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.5.1.36 sKSCW

Definition:

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	opCd	UBYTE	operation code
	bsTp	UBYTE	basic service type
	bsCd	UBYTE	basic service code

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.5.1.37 sKSCL

Definition:

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	opCd	UBYTE	operation code
	ssCd	UBYTE	supplementary service code

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.5.1.38 sKSPW**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	ssCd	UBYTE	supplementary service code
	oldPwd	UBYTE*	old password
	newPwd	UBYTE*	new password

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.5.1.39 sKSUB**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	ssCd	UBYTE	supplementary service code
	puk	UBYTE*	PUK
	pin	UBYTE*	PIN

4.4.5.1.40 sKSUS**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source identifier
	ussd	UBYTE*	USSD string

4.4.5.1.41 sPVRF**Definition:**

struct type	element	element type	comment
	srcId	T_ACI_CMD_SRC	command source
	type	T_ACI_PVRF_TYPE	type of pin (PINn/PUKn)
	pin	CHAR*	PIN
	newpin	CHAR*	new PIN

4.4.6 T_ACI_CLSSTAT**Type Declaration:**

struct type	element	element type	comment
T_ACI_CLSSTAT	status	T_ACI_STATUS	status information
	class	T_ACI_CLASS	class information

4.4.7 T_ACI_CMD_DATA**Type Declaration:**

struct type	element	element type	comment
T_ACI_CMD_DATA	data	UBYTE[MAX_SM_CMD_LEN]	SMS command data

struct type	element	element type	comment
	len	UBYTE	SMS command data length

4.4.8 T_ACI_CMGL_CBM

Type Declaration:

struct type	element	element type	comment
T_ACI_CMGL_CBM	index	SHORT	value in the range of location numbers supported by the associated memory
	stat	T_ACI_SMS_STAT	status of message in memory
	sn	USHORT	CBM serial number
	mid	USHORT	CBM message identifier
	page	UBYTE	CBM page parameter bits 4-7
	pages	UBYTE	CBM page parameter bits 0-3
	data	T_ACI_CBM_DATA	message data

4.4.9 T_ACI_CMGL_SM

Type Declaration:

struct type	element	element type	comment
T_ACI_CMGL_SM	index	SHORT	value in the range of location numbers supported by the associated memory
	stat	T_ACI_SMS_STAT	status of message in memory
	adress	CHAR[MAX_NUM_LEN]	originating/destination address
	alpha	T_ACI_PB_TEXT	alphanumeric entry corresponding to phone number
	scts	T_ACI_VP_ABS	service center time stamp
	toa	T_ACI_TOA	<oa/da> type of address
	data	T_ACI_SM_DATA	message data

4.4.10 T_ACI_CMGR_CBM

Type Declaration:

struct type	element	element type	comment
T_ACI_CMGR_CBM	stat	T_ACI_SMS_STAT	status of message in memory
	sn	USHORT	CBM serial number
	mid	USHORT	CBM message identifier
	dcs	UBYTE	SMS/CB Data Coding Scheme
	page	UBYTE	CBM page parameter bits 4-7
	pages	UBYTE	CBM page parameter bits 0-3
	data	T_ACI_CBM_DATA	message data

4.4.11 T_ACI_CMGR_SM

Type Declaration:

struct type	element	element type	comment
T_ACI_CMGR_SM	stat	T_ACI_SMS_STAT	status of message in memory
	addr	CHAR[MAX_NUM_LEN]	originator/destination address
	toa	T_ACI_TOA	type of originator/destination address
	sca	CHAR[MAX_NUM_LEN]	RP SC address Address-Value field in string format
	tosca	T_ACI_TOA	type of service center address
	alpha	T_ACI_PB_TEXT	alphanumeric entry corresponding to phone number
	vprel	UBYTE	relative validity period
	vpabs_scts	T_ACI_VP_ABS	absolute validity period or service center time stamp
	fo	UBYTE	first octet of SMS-DELIVER, SMS-SUBMIT, SMS-STATUS-REPORT or SMS-COMMAND
	pid	UBYTE	TP Protocol Identifier
	dcs	UBYTE	SMS/CB Data Coding Scheme
	data	T_ACI_SM_DATA	message data
	udh	T_ACI_UDH_DATA	user data header

4.4.12 T_ACI_CNUM_MSISDN

Type Declaration:

struct type	element	element type	comment
T_ACI_CNUM_MSISDN	alpha	CHAR[MAX_ALPHA_LEN]	optional alphanumeric string associated with <number>; character set used should be the one selected with command Select TE Character Set +CSCS
	number	CHAR[MAX_NUM_LEN]	string type phone number of format specified by <type>
	type	T_ACI_TOA	type of address
	speed	T_ACI_BS_SPEED	data rate of bearer service
	service	T_ACI_CNUM_SERV	service related to the phone number
	itc	T_ACI_CNUM_ITC	information transfer capability

4.4.13 T_ACI_COPN_OPDESC

Type Declaration:

struct type	element	element type	comment
T_ACI_COPN_OPDESC	alphaOper	CHAR[MAX_ALPHA_OPER_LEN]	operator in alphanumeric format
	numOper	CHAR[MAX_NUM_OPER_LEN]	operator in numeric format

4.4.14 T_ACI_COPS_OPDESC

Type Declaration:

struct type	element	element type	comment
-------------	---------	--------------	---------

struct type	element	element type	comment
T_ACI_COPS_OPDESC	status	T_ACI_COPS_STAT	indicating the availability of the operator
	longOper	CHAR*	long alphanumeric format of the name of the operator
	shortOper	CHAR*	short alphanumeric format of the name of the operator
	numOper	CHAR[MAX_NUM_OPER_LEN]	numeric format representation of the operator

4.4.15 T_ACI_CPOL_OPDESC

Type Declaration:

struct type	element	element type	comment
T_ACI_CPOL_OPDESC	index	SHORT	index, order number of operator in the SIM preferred operator list
	format	T_ACI_CPOL_FRMT	format of oper
	oper	CHAR[MAX_ALPHA_OPER_LEN]	network operator

4.4.16 T_ACI_KSIR

Type Declaration:

struct type	element	element type	comment
T_ACI_KSIR	ksdCmd	T_ACI_KSD_CMD	key sequence decoder command, defines the used structure of the union <ir>
	ir	union	result union

4.4.16.1 ir

Definition:

union type	element	element type	comment
	rKSCL	struct	calling line
	rKSCW	struct	call waiting
	rKSCF	struct	call forwarding
	rKSCB	struct	call blocking
	rKSPW	struct	password
	rkSUS	struct	supplementary service
	rKSIMEI	struct	IMEI

4.4.16.1.1 rKSCL

Definition:

struct type	element	element type	comment
	opCd	UBYTE	operation code
	ssCd	UBYTE	supplementary service code
	ssErr	UBYTE	supplementary service error
	ssSt	UBYTE	supplementary service status

struct type	element	element type	comment
	clirOpt	UBYTE	CLIR option
	ovrCtg	UBYTE	override category

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.16.1.2 rKSCW

Definition:

struct type	element	element type	comment
	opCd	UBYTE	operation code
	ssCd	UBYTE	supplementary service code
	ssErr	UBYTE	supplementary service error
	ssSt	UBYTE	supplementary service state
	c_cwBSGLst	UBYTE	call waiting basic service group count
	cwBSGLst	T_Cx_BSG*	call waiting basic service group list

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.16.1.2.1 T_Cx_BSG

Type Declaration:

array type	element type	element count	comment
T_Cx_BSG	UBYTE	bsTp	basic service type
	UBYTE	bsCd	basic service code

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.16.1.3 rKSCF

Definition:

struct type	element	element type	comment
	opCd	UBYTE	operation code
	ssCd	UBYTE	supplementary service code
	ssErr	UBYTE	supplementary service error
	c_cfFeatLst	UBYTE	call forwarding feature count
	cfFeatLst	T_CF_FEAT*	call forwarding feature list

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.16.1.3.1 T_CF_FEAT

Type Declaration:

array type	element type	element count	comment
T_CF_FEAT	UBYTE	bsTp	basic service type
	UBYTE	bsCd	basic service code
	UBYTE	ssSt	supplementary service state
	UBYTE[MAX_NUM_LEN]	num	number

UBYTE	ton	type of number
UBYTE	npi	numbering plan
UBYTE[MAX_SUBADDR_LEN]	sub	subaddress
UBYTE	tos	type of subaddress
UBYTE	oe	odd/even indicator
UBYTE	time	no reply time

4.4.16.1.4 rKSCB

Definition:

struct type	element	element type	comment
	opCd	UBYTE	operation code
	ssCd	UBYTE	supplementary service code
	ssErr	UBYTE	supplementary service error
	c_cbInfoLst	UBYTE	call barring info count
	cbInfoLst	T_CB_INFO*	call barring info list

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.16.1.4.1 T_CB_INFO

Type Declaration:

array type	element type	element count	comment
T_CB_INFO	UBYTE	bsTp	basic service type
	UBYTE	bsCd	basic service code
	UBYTE	ssSt	supplementary service state

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.16.1.5 rKSPW

Definition:

struct type	element	element type	comment
	opCd	UBYTE	operation code
	ssCd	UBYTE	supplementary service code
	ssErr	UBYTE	supplementary service error
	newPwd	UBYTE[MAX_FAC_PWD_LEN]	new password

For further details of some elements of the structure, see recommendation GSM 04.80.

4.4.16.1.6 rKSUS

Definition:

struct type	element	element type	comment
	ussd	UBYTE*	ussd string

4.4.16.1.7 rKSIMEI

Definition:

struct type	element	element type	comment
	UBYTE	tac1	type approval code
	UBYTE	tac2	type approval code
	UBYTE	tac3	type approval code
	UBYTE	fac	final assembly code
	UBYTE	snr1	serial number
	UBYTE	snr2	serial number
	UBYTE	snr3	serial number
	UBYTE	svn	software version number

4.4.17 T_ACI_PB_ENTR

Type Declaration:

struct type	element	element type	comment
T_ACI_PB_ENTR	index	SHORT	value in the range of location numbers of phone book memory
	number	CHAR[MAX_NUM_LEN]	phone number of format <type>
	type	T_ACI_TOA	type of address
	text	T_ACI_PB_TEXT	phonebook text

4.4.18 T_ACI_PB_TEXT

Type Declaration:

struct type	element	element type	comment
T_ACI_PB_TEXT	data	UBYTE [MAX_ALPHA_LEN]	phonebook text data
	len	UBYTE	length of phonebook text data

4.4.19 T_ACI_RLOG

Type Declaration:

struct type	element	element type	comment
T_ACI_RLOG	atRslt	T_ACI_RLOG_RSLT	result of AT command, used to select one of the rXX structure elements in the union rsltPrm
	dest	T_ACI_CMD_SRC	command source
	rsltPrm	union	structure to give further information about success or failure of AT command

4.4.19.1 rsltPrm

Definition:

union type	element	element type	comment
	rOk	struct	used if "OK"
	rNO_CARRIER	struct	used if "NO CARRIER"
	rCONNECT	struct	user if "CONNECT"
	rBUSY	struct	used if "BUSY"

union type	element	element type	comment
	rNO_ANSWER	struct	used if "NO ANSWER"
	rCME	struct	user if "ERROR"

4.4.19.1.1 rOK**Definition:**

struct type	element	element type	comment
	cmdId	T_ACI_CMD	command source
	cId	SHORT	call identifier

4.4.19.1.2 rNO_CARRIER**Definition:**

struct type	element	element type	comment
	cmdId	T_ACI_CMD	command source
	cId	SHORT	call identifier

4.4.19.1.3 rCONNECT**Definition:**

struct type	element	element type	comment
	cmdId	T_ACI_CMD	command source
	speed	T_ACI_BS_SPEED	connection rate
	cId	SHORT	call identifier

4.4.19.1.4 rBUSY**Definition:**

struct type	element	element type	comment
	cmdId	T_ACI_CMD	command source
	cId	SHORT	call identifier

4.4.19.1.5 rNO_ANSWER**Definition:**

struct type	element	element type	comment
	cmdId	T_ACI_CMD	command source
	cId	SHORT	call identifier

4.4.19.1.6 rCME**Definition:**

struct type	element	element type	comment
	cmdId	T_ACI_CMD	command source
	err	T_ACI_CME_ERR	error description

4.4.20 T_ACI_RPCT_VAL**Type Declaration:**

struct type	element	element type	comment
T_ACI_RPCT_VAL	currency	UBYTE[MAX_CUR_LEN]	currency
	eppu	ULONG	elementary price per unit
	exp	ULONG	expression value
	sexp	ULONG	sign of expression

4.4.21 T_ACI_SM_DATA**Type Declaration:**

struct type	element	element type	comment
T_ACI_SM_DATA	data	UBYTE [MAX_SM_LEN]	short message data
	len	UBYTE	short message data length

4.4.22 T_ACI_SMS_STOR_OCC**Type Declaration:**

struct type	element	element type	comment
T_ACI_SMS_STOR_OCC	mem	T_ACI_SMS_STOR	memory for storing messages
	used	SHORT	number of messages currently in <mem>
	total	SHORT	total number of message locations in <mem>

4.4.23 T_ACI_TOA**Type Declaration:**

struct type	element	element type	comment
T_ACI_TOA	ton	T_ACI_TOA_TON	type of number
	npi	T_ACI_TOA_NPI	numbering plan identification

4.4.24 T_ACI_TOS**Type Declaration:**

struct type	element	element type	comment
T_ACI_TOS	tos	T_ACI_TOS_TOS	type of subaddress
	oe	T_ACI_TOS_OE	odd/even indicator

4.4.25 T_ACI_UDH_DATA**Type Declaration:**

struct type	element	element type	comment
T_ACI_UDH_DATA	data	UBYTE[MAX_SM_LEN]	UDH data
	len	UBYTE	UDH data length

4.4.26 T_ACI_USSD_DATA**Type Declaration:**

struct type	element	element type	comment
T_ACI_USSD_DATA	data	UBYTE[MAX_USSD_LEN]	USSD data
	len	UBYTE	USSD data length

4.4.27 T_ACI_VP_ABS**Type Declaration:**

struct type	element	element type	comment
T_ACI_VP_ABS	year	UBYTE[MAX_VP_ABS_DIGITS]	year in BCD
	month	UBYTE[MAX_VP_ABS_DIGITS]	month in BCD
	day	UBYTE[MAX_VP_ABS_DIGITS]	day in BCD
	hour	UBYTE[MAX_VP_ABS_DIGITS]	hour in BCD
	minute	UBYTE[MAX_VP_ABS_DIGITS]	minute in BCD
	second	UBYTE[MAX_VP_ABS_DIGITS]	second in BCD
	timezone	SHORT	timezone in quarters of hours

4.5 Arrays**4.5.1 T_ACI_CAL_LST****Type Declaration:**

array type	element type	element count	comment
T_ACI_CAL_LST	T_ACI_CAL_ENTR	MAX_CALL_NR	a list of all current calls managed by the ME

4.5.2 T_ACI_CCFC_LST**Type Declaration:**

array type	element type	element count	comment
T_ACI_CCFC_LST	T_ACI_CCFC_SET	MAX_CLASS	list of call forwarding settings

4.5.3 T_ACI_CLCC_LST**Type Declaration:**

array type	element type	element count	comment
T_ACI_CLCC_LST	T_ACI_CLCC_CALDESC	MAX_CALL_NR	list of call descriptions

4.5.4 T_ACI_CLSSTAT_LST**Type Declaration:**

array type	element type	element count	comment
T_ACI_CLSSTAT_LST	T_ACI_CLSSTAT	MAX_CLASS	list of class/status

4.5.5 T_ACI_CMGL_CBM_LST**Type Declaration:**

array type	element type	element count	comment
T_ACI_CMGL_CBM_LST	T_ACI_CMGL_CBM	MAX_CBM_ENTR	a list of messages with status value <stat> from preferred message storage <mem1>

4.5.6 T_ACI_CMGL_SM_LST

Type Declaration:

array type	element type	element count	comment
T_ACI_CMGL_SM_LST	T_ACI_CMGL_SM	MAX_SM_ENTR	a list of messages with status value <stat> from preferred message storage <mem1>

4.5.7 T_ACI_CNUM_LST

Type Declaration:

array type	element type	element count	comment
T_ACI_CNUM_LST	T_ACI_CNUM_MSISDN	MAX_MSISDN	a list of MSISDNs related to the subscriber

4.5.8 T_ACI_COPS_LST

Type Declaration:

array type	element type	element count	comment
T_ACI_COPS_LST	T_ACI_COPS_OPDESC	MAX_OPER	a list of operators present in network

4.5.9 T_ACI_PB_LST

Type Declaration:

array type	element type	element count	comment
T_ACI_PB_LST	T_ACI_PB_ENTR	MAX_PB_ENTR	List of phonebook entries

5 Constants

5.1 General

This section describes the constants required to drive the functional interface of the ACI.

There is one table containing all of the information related to the specific constants. The summary below explains the individual columns in detail:

Column Header	Description
Name	name of the constant
Value	Numerical value represented by the constant
Comment	Description of the purpose of the constant

5.2 Definitions

name	value	comment
ACI_NumParmNotPresent	-1	numerical parameter not present
MAX_OPER	5	maximum number of operators in operator list
MAX_CLASS	7	maximum number of classes corresponding to the enum type T_ACI_CLCK_CLS
MAX_FACILITY	17	maximum number of supported facilities
MAX_CALL_NR	4	maximum number of current calls managed by the ME
MAX_NUM_LEN	21	maximum length of a phone number including null termination
MAX_SUBADDR_LEN	21	maximum length of a subaddress including null termination
MAX_ALPHA_LEN	21	maximum length of an alphanumerical phone book entry including null termination
MAX_ALPHA_OPER_LEN	21	maximum length of the operator name in alphanumerical format including null termination
MAX_NUM_OPER_LEN	7	maximum length of the operator name in numerical format including null termination
MAX_VP_ABS_DIGITS	2	maximum number of BCD digits in absolute time
MAX_SM_LEN	161	maximum length of a short message
MAX_CBM_LEN	94	maximum length of a cell broadcast message
MAX_SM_CMD_LEN	158	maximum length of a short message command
MAX_CBM_TYPES	20	maximum number of data coding scheme or messages identifier ranges
MAX_PB_ENTR	5	maximum number of phone book entries in list
MAX_PB_INDEX	255	maximum phone book index
MAX_SM_ENTR	1	maximum number of entries in SM storage
MAX_FAC_PWD_LEN	5	maximum facility password length
MAX_USSD_LEN	161	maximum length of a ussd message
MAX_DIAL_LEN	161	maximum dial string length
MAX_IMSI_LEN	18	maximum length of an IMSI
MAX_SAT_PRF_LEN	MAX_STK_PRF	maximum SIM application toolkit profile length

MAX_SAT_CMD_LEN	255	maximum SIM application toolkit command length
MAX_CUR_LEN	4	maximal currency length
MAX_MSISDN	2	maximum number of MSISDNs related to the subscriber
CBCH_HEAD_LEN	6	cell broadcast channel header length
CMGW_IDX_FREE_ENTRY	0	write message to next free mem location instead given index
CMGL_RESTART_IDX	1	restart reading of SMS list from SIM and ME memory
MAX_CBM_ENTR	xxx	maximum number of entries in CBM storage

6 Functions Derived from GSM Rec. 07.07

6.1 General Commands

6.1.1 Set Functions

6.1.1.1 sAT_PlusWS46 – Select Wireless Network

Command Reference:

GSM 07.07 (5.9)

Function Definition:

T_ACI_RETURN sAT_PlusWS46 (T_ACI_CMD_SRC srcId, T_ACI_WS46_MOD mode);

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
mode	---	wireless network	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

General Commands

Description:

The function sAT_PlusWS46 () selects the cellular network given by the value *mode* to operate with the TA.

Related Functions:

qAT_PlusWS46 ()

6.1.2 Query Functions

6.1.2.1 qAT_PlusCIMI () – Request International Mobile Subscriber Identity

Command Reference:

GSM 07.07 (5.6)

Function Definition:

T_ACI_RETURN qAT_PlusCIMI (T_ACI_CMD_SRC srcId, CHAR* imsi);

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
imsi	MAX_IMSI_LEN	international mobile subscriber identity	OUT

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

General Commands

Description:

The function qAt_PlusCIMI queries the international mobile subscriber identity. The returned value is stored in the buffer provided by the value *imsi* if the function returns successfully. If the processing of the request can not be completed immediately, the function returns with the value AT_EXCT, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCIMI.

Related Functions:

rAT_PlusCIMI ()

rAT_PlusCME ()

rAT_OK ()

6.1.2.2 qAT_PlusWS46 – Select Wireless Network

Command Reference:

GSM 07.07 (5.9)

Function Definition:

```
T_ACI_RETURN qAT_PlusWS46 (T_ACI_CMD_SRC srcId, T_ACI_WS46_MOD* mode);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
mode	---	wireless network	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

General Commands

Description:

The function qAT_PlusWS46 () returns the currently selected cellular network in the location given by the value *mode*.

Related Functions:

sAT_PlusWS46 ()

6.1.3 Call-back Functions

6.1.3.1 rAT_PlusCIMI () – International Mobile Subscriber Identity

Command Reference:

GSM 07.07 (5.6)

Function Definition:

```
void rAT_PlusCIMI (CHAR* imsi);
```

Parameters:

name	buffer size	comment	
imsi	MAX_IMSI_LEN	international mobile subscriber identity	IN

Return:

none

Function Group:

General Commands

Description:

The function rAt_PlusCIMI informs the application about the international mobile subscriber identity. *imsi* is the pointer to the null terminated IMSI string.

Related Functions:

qAT_PlusCIMI ()

6.2 Call Control Functions

6.2.1 Set Functions

6.2.1.1 sAT_PlusCSTA () - Select Type of Address

Command Reference:

GSM 07.07 (6.1)

Function Definition:

T_ACI_RETURN sAT_PlusCSTA (T_ACI_CMD_SRC srcId, T_ACI_TOA* type);

Parameters:

name	comment	
srcId	command source identifier	IN
type	type of address	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Call Control

Description:

The sAT_PlusCSTA function selects the type of number for further dialing (sAT_Dn, sAT_Dm) if it returns successfully. The value of *type* represents the type of address.

Related Functions:

qAT_PlusCSTA ()

6.2.1.2 sAT_Dn () - Originate Call to Phone Number Provided

Command Reference:

GSM 07.07 (6.2)

Function Definition:

```
T_ACI_RETURN sAT_Dn (T_ACI_CMD_SRC srcId, CHAR* number, T_ACI_D_CLIR_OVRD clirOvr,
                     T_ACI_D_CUG_CTRL cugCtrl, T_ACI_D_TOC callType);
```

Parameters:

name	comment	
srcId	command source identifier	IN
number	phone number	IN
clirOvr	overrides the CLIR supplementary service subscription default value for this call	IN
cugCtrl	controls the CUG supplementary service information for this call	IN
callType	sets whether or not a voice call is initiated	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Call Control

Description:

The sAT_Dn function causes the ME to establish a call to the phone number stored in the buffer provided by *number*. The dial string consists of a combination of the following digits: 1234567890*#+ABC. All other characters in the dial string are ignored.

The parameters *clirOvr*, *cugCtrl* and *callType* specify the characteristics for the call setup. *ClirOvr* overrides the CLIR supplementary service subscription default value for this call. *CugCtrl* controls the CUG supplementary service information for this call while the index and info values set with the sAT_PlusCCUG function are used. *CallType* defines the type of call, whether it is a voice or data call. In case of a data call also the settings of sAT_PlusCBST and sAT_plusCMOD are relevant for the call set up.

Related Functions:

rAT_OK ()
rAT_NO_CARRIER()

6.2.1.3 sAT_Dm () - Originate a Call Using Phonebook Memory**Command Reference:**

GSM 07.07 (6.3)

Function Definition:

```
T_ACI_RETURN sAT_Dm (T_ACI_CMD_SRC srcId, CHAR* str, T_ACI_PB_STOR mem, SHORT index,
                     T_ACI_D_CLIR_OVRD clirOvr, T_ACI_D_CUG_CTRL cugCtrl,
                     T_ACI_D_TOC callType);
```

Parameters:

name	comment	
srcId	command source identifier	IN
str	requested entry to dial	IN
mem	phonebook memory	IN
index	phonebook memory entry location	IN
clirOvr	overrides the CLIR supplementary service subscription default value for this call	IN
cugCtrl	controls the CUG supplementary service information for this call	IN
callType	sets whether or not a voice call is initiated	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Call Control

Description:

The sAT_Dm function causes the ME to establish a call to a phone number stored in the phonebook. The phonebook entry is either specified by the memory *mem* and/or the entry location *index* or the alphanumeric field stored in the buffer provided by *str*. Only one of the following combinations is supported while all others result in a failure:

str - present, *mem* - not present, *index* - not present:

originate call to phone number of which the corresponding alphanumeric field is *str* (if possible, all available memories should be searched for the correct entry),

str - not present, *mem* - present, *index* - present:

originate call to phone number in memory *mem* entry location *index* (available memories may be queried with Select Phonebook Storage query function qAT_PlusCPBS; *mem* could be e.g. ME or SIM),

str - not present, *mem* - not present, *index* - present:

originate call to phone number in entry location *index* (it is manufacturer-specific which memory storage of ME, SIM and TA is used; function Select Phonebook Memory Storage sAT_PlusCPBS setting is recommended).

The parameters *clirOvr*, *cugCtrl* and *callType* specify the characteristics for the call setup. *ClirOvr* overrides the CLIR supplementary service subscription default value for this call. *CugCtrl* controls the CUG supplementary service information for this call while the index and info values set with the sAT_PlusCCUG function are used. *CallType* defines the type of call, whether it is a voice or data call. In case of a data call also the settings of sAT_PlusCBST and sAT_plusCMOD are relevant for the call set up.

Related Functions:

rAT_OK ()
rAT_NO_CARRIER()

6.2.1.4 sAT_PlusCMOD () - Call Mode

Command Reference:

GSM 07.07 (6.4)

Function Definition:

```
T_ACI_RETURN sAT_PlusCMOD (T_ACI_CMD_SRC srcId, T_ACI_CMOD_MOD mode);
```

Parameters:

name	comment	
srcId	command source identifier	IN
mode	call mode	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Call Control

Description:

The sAT_PlusCMOD function selects the call mode of further dialing (sAT_Dn, sAT_Dm) or for the next answering (sAT_A) if it returns successfully. The value of *mode* represents the call mode. The value of *mode* is reset automatically to single mode after successful completion of the call or an unsuccessful answering.

Related Functions:

qAT_PlusCMOD ()

6.2.1.5 sAT_PlusCHUP () - Hangup Call

Command Reference:

GSM 07.07 (6.5)

Function Definition:

```
T_ACI_RETURN sAT_PlusCHUP (T_ACI_CMD_SRC srcId);
```

Parameters:

name	comment	
srcId	command source identifier	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Call Control

Description:

The sAT_PlusCHUP function causes the ME to hang-up the current call. The purpose of this function is not to substitute the function sAT_H, but to give an assured procedure for terminating an alternating mode call.

Related Functions:

rAT_OK ()

6.2.1.6 sAT_PlusCBST () - Select Bearer Service Type

Command Reference:

GSM 07.07 (6.7)

Function Definition:

```
T_ACI_RETURN sAT_PlusCBST (T_ACI_CMD_SRC srcId, T_ACI_BS_SPEED speed, T_ACI_CBST_NAM name,  
                           T_ACI_CBST_CE ce);
```

Parameters:

name	comment	
srcId	command source identifier	IN
speed	data rate	IN
name	bearer service	IN
ce	connection element	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Call Control

Description:

The sAT_PlusCBST function selects a bearer service with a specific data rate and a specific connection element to be used when data calls are originated if it returns successfully. The value of *name* represents the bearer service, the value of *speed* - the data rate and the value of *ce* - the connection element.

Related Functions:

qAT_PlusCBST ()

6.2.1.7 sAT_PlusCRLP () Radio Link Protocol

Command Reference:

GSM 07.07 (6.8)

Function Definition:

T_ACI_RETURN sAT_PlusCRLP (T_ACI_CMD_SRC srcId, SHORT iws, SHORT mws, SHORT t1, SHORT n2);

Parameters:

name	comment	
srcId	command source identifier	IN
iws	IWF to MS window size	IN
mws	MS to IWF window	IN
t1	acknowledgement timer	IN
n2	retransmission attempts	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Call Control

Description:

The sAT_PlusCRLP function is used to alter the radio link protocol parameters. The value of *iws* represents the IWF to MS window size with a default value of 61, the value of *mws* the MS to IWF window size with a default value of 61, the value of *t1* the acknowledgement timer with a default value of 48 and the value of *n2* the number of retransmission attempts with a default value of 6.

Related Functions:

qAT_PlusCRLP ()

6.2.1.8 sAT_PlusCSNS () Single Numbering Scheme

Command Reference:

GSM 07.07 (6.17)

Function Definition:

```
T_ACI_RETURN sAT_PlusCSNS (T_ACI_CMD_SRC srcId, T_ACI_CSNS_MOD mode);
```

Parameters:

name	comment	
srcId	command source identifier	IN
mode	bearer or teleservice mode	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Call Control

Description:

The sAT_PlusCSNS function selects the bearer or teleservice to be used when mobile terminated single numbering scheme call is established. The value given by *mode* selects the respective bearer or teleservice and the behaviour of the ME when accepting the call.

Related Functions:

qAT_PlusCSNS ()

6.2.2 Query Functions

6.2.2.1 qAT_PlusCSTA () - Select Type of Address

Command Reference:

GSM 07.07 (6.1)

Function Definition:

T_ACI_RETURN qAT_PlusCSTA (T_ACI_CMD_SRC srcId, T_ACI_TOA* type);

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
type	---	type of address	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Call Control

Description:

The qAT_PlusCSTA function returns the current setting for type of number and numbering plan which is used for further dialing functions according to GSM specifications. The return value is the type of address (refer to GSM 04.08 [8] subclause 10.5.4.7) which is stored in the location provided by *type* if the function returns successfully.

Related Functions:

sAT_PlusCSTA ()

6.2.2.2 qAT_PlusCMOD () - Call Mode

Command Reference:

GSM 07.07 (6.4)

Function Definition:

```
T_ACI_RETURN qAT_PlusCMOD (T_ACI_CMD_SRC srcId, T_ACI_CMOD_MOD* mode);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
mode	---	call mode	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Call Control

Description:

The qAT_PlusCMOD function returns the current setting for the call mode. The return value is stored in the location provided by *mode* if the function returns successfully.

Related Functions:

sAT_PlusCMOD ()

6.2.2.3 qAT_PlusCBST () - Bearer Service Type Selection

Command Reference:

GSM 07.07 (6.7)

Function Definition:

```
T_ACI_RETURN qAT_PlusCBST (T_ACI_CMD_SRC srcId, T_ACI_BS_SPEED* speed,  
                           T_ACI_CBST_NAM* name, T_ACI_CBST_CE* ce);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
speed	---	data rate	OUT
name	---	bearer service	OUT
ce	---	connection element	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Call Control

Description:

The qAT_PlusCBST function returns the current setting of the bearer service type, including data rate, service and connection element. The return values are stored in the respective locations provided by *speed*, *name* and *ce* if the function returns successfully.

Related Functions:

sAT_PlusCBST ()

6.2.2.4 qAT_PlusCRLP () Radio Link Protocol

Command Reference:

GSM 07.07 (6.8)

Function Definition:

```
T_ACI_RETURN qAT_PlusCRLP (T_ACI_CMD_SRC srcId, SHORT* iws, SHORT* mws, SHORT* t1,  
                           SHORT* n2);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
iws	---	IWF to MS window size	OUT
mws	---	MS to IWF window size	OUT
t1	---	acknowledge timer	OUT
n2	---	retransmission attempts	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Call Control

Description:

The qAT_PlusCRLP function returns the current setting of the parameters defining the Radio Link Protocol, including the number of outstanding frames for the up-link and down-link directions, the acknowledge timer and the number of retransmission attempts. The return values are stored in the respective locations provided by *iws*, *mws*, *t1* and *n2* if the function returns successfully.

Related Functions:

sAT_PlusCRLP ()

6.2.2.5 qAT_PlusCEER () – Extended Error Report

Command Reference:

GSM 07.07 (6.10)

Function Definition:

T_ACI_RETURN qAT_PlusCEER (T_ACI_CMD_SRC srcId, T_ACI_CEER* report);

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
report	---	error report	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Call Control

Description:

The qAT_PlusCEER function returns in the location provided by *report* the reason for the last unsuccessful call setup, in-call modification or the reason for the last call release.

Related Functions:

6.2.2.6 qAT_PlusCSNS () – Single Numbering Scheme

Command Reference:

GSM 07.07 (6.17)

Function Definition:

```
T_ACI_RETURN qAT_PlusCSNS (T_ACI_CMD_SRC srcId, T_ACI_CSNS_MOD* mode);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
mode	---	bearer or teleservice mode	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Call Control

Description:

The qAT_PlusCSNS function returns in the location provided by *mode* the bearer or teleservice used by mobile terminated calls if single numbering scheme is used.

Related Functions:

sATplusCSNS ()

6.2.3 Call-back Functions

6.2.3.1 rAT_PlusCR () - Service Reporting Control

Command Reference:

GSM 07.07 (6.9)

Function Definition:

```
void rAT_PlusCR (T_ACI_CR_SERV service);
```

Parameters:

name	comment	
service	used service	IN

Return:

none

Function Group:

Results

Description:

The rAT_PLUSCR function is used to report the speed and quality for a connection. The negotiated service for the connection is passed via *service* before a connect report is made.

Related Functions:

6.2.3.2 rAT_PlusCRING () - Incoming Call Indication

Command Reference:

GSM 07.07 (6.11)

Function Definition:

```
void rAT_PlusCRING (T_ACI_CRING_MOD mode, T_ACI_CRING_TYP type1, T_ACI_CRING_TYP type2);
```

Parameters:

name	comment	
mode	mode of call sequence	IN
type1	type of the call 1	IN
type2	type of the call 2	IN

Return:

none

Function Group:

Results

Description:

The rAT_PlusCRING function is used to report an incoming call. The more detailed information about the type of the call is indicated via *type1* and *type2*.

Related Functions:

rAT_PlusCRING_OFF ()

6.2.3.3 rAT_PlusCRING_OFF () - Incoming Call Indication OFF

Command Reference:

GSM 07.07 (6.11)

Function Definition:

```
void rAT_PlusCRING (SHORT cId);
```

Parameters:

name	comment	
cId	call Id	IN

Return:

none

Function Group:

Results

Description:

The rAT_PlusCRING_OFF function is used to indicate that an incoming call is no longer waiting. The value *cId* represents the call number. A call is no longer waiting when: the call was accepted, the call was rejected or the other party has disconnected the call while the call was still pending.

Related Functions:

rAT_PlusCRING ()

6.3 Network Service Related Functions

1. Set Functions

6.3.1.1 sAT_PlusCOPS () - Operator Selection

Command Reference:

GSM 07.07 (7.3)

Function Definition:

T_ACI_RETURN sAT_PlusCOPS (T_ACI_CMD_SRC srcId, T_ACI_COPS_MOD mode,
T_ACI_COPS_FRMT format, CHAR* oper);

Parameters:

name	comment	
srcId	command source identifier	IN
mode	selection mode	IN
format	format of the operator name	IN
oper	operator name	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Network

Description:

The sAT_PlusCOPS function forces an attempt to select and register the GSM network operator. The parameter *mode* is used to select whether the selection is done automatically by the ME or forced by this function to the operator whose name is stored in the buffer provided by *oper* (the value of *oper* is provided in format *format*).

Related Functions:

qAT_PlusCOPS ()
rAT_PlusCOPS ()
tAT_PlusCOPS ()
rAT_PlusCME ()
rAT_OK ()

6.3.1.2 sAT_PlusCLCK () - Facility Lock

Command Reference:

GSM 07.07 (7.4)

Function Definition:

```
T_ACI_RETURN sAT_PlusCLCK (T_ACI_CMD_SRC srcId, T_ACI_CLCK_FAC fac, T_ACI_CLCK_MOD mode,  
                           CHAR* passwd, T_ACI_CLASS class);
```

Parameters:

name	comment	
srcId	command source identifier	IN
fac	ME or network facility	IN
mode	locking mode	IN
passwd	ME or network facility password	IN
class	service class	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The sAT_PlusCLCK function is used to lock or unlock an ME or network facility. A password is normally required to perform such actions. The value of *fac* represents the specific facility, the value provided by *mode* - the lock condition and the buffer provided by *passwd* contains the same character sequence as specified for the facility with function sAT_PlusCPWD.

Related Functions:

qAT_PlusCLCK ()
rAT_PlusCLCK ()
rAT_PlusCME ()
rAT_OK ()

6.3.1.3 sAT_PlusCPWD () - Change Password

Command Reference:

GSM 07.07 (7.5)

Function Definition:

```
T_ACI_RETURN sAT_PlusCPWD (T_ACI_CMD_SRC srcId, T_ACI_CPWD_FAC fac, CHAR* oldpwd,
                           CHAR* newpwd);
```

Parameters:

name	comment	
srcId	command source identifier	IN
fac	ME or network facility	IN
oldpwd	old password for ME or network facility	IN
newpwd	new password for ME or network facility	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The sAT_PlusCPWD function is used to set or change a password for the facility lock functionality defined by function sAT_PlusCLCK. The value of *fac* represents the specific facility. The buffers provided by *oldpwd* and *newpwd* store the old and new passwords.

Related Functions:

tAT_PlusCPWD ()

rAT_PlusCPWD ()

rAT_PlusCME ()

rAT_OK ()

6.3.1.4 sAT_PlusCLIR () - Calling Line Identification Restriction**Command Reference:**

GSM 07.07 (7.7)

Function Definition:

```
T_ACI_RETURN sAT_PlusCLIR (T_ACI_CMD_SRC srcId, T_ACI_CLIR_MOD mode);
```

Parameters:

name	comment	
srcId	command source identifier	IN
mode	CLIR mode	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Supplementary Services

Description:

The sAT_PlusCLIR function is used to control whether or not the presentation of the calling line identity to the called party is allowed when originating a call. The value of *mode* represents whether this service is invoked or suppressed.

Related Functions:

qAT_PlusCLIR ()

rAT_PlusCLIR ()

rAT_PlusCME ()

rAT_OK ()

6.3.1.5 sAT_PlusCCUG () - Closed User Group

Command Reference:

GSM 07.07 (7.9)

Function Definition:

T_ACI_RETURN sAT_PlusCCUG (T_ACI_CMD_SRC srcId, T_ACI_CCUG_MOD mode, T_ACI_CCUG_IDX index, T_ACI_CCUG_INFO info);

Parameters:

name	comment	
srcId	command source identifier	IN
mode	CUG mode	IN
index	CUG index	IN
info	suppression information	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Supplementary Services

Description:

The sAT_PlusCCUG function controls the closed user group service. The *mode* value represents whether or not CUG temporary mode is enabled or disabled, the *index* value - the CUG index and the *info* value - whether or not OA and/or preferential CUG is suppressed.

Related Functions:

qAT_PlusCCUG ()

6.3.1.6 sAT_PlusCCFC () - Call Forwarding Number and Conditions**Command Reference:**

GSM 07.07 (7.10)

Function Definition:

T_ACI_RETURN sAT_PlusCCFC (T_ACI_CMD_SRC srcId, T_ACI_CCFC_RSN reason, T_ACI_CCFC_MOD mode, CHAR* number, T_ACI_TOA* type, T_ACI_CLASS class, CHAR* subaddr, T_ACI_TOS* satype, SHORT time);

Parameters:

name	comment	
srcId	command source identifier	IN
reason	forwarding reason	IN
mode	forwarding mode	IN
number	forwarding address	IN
type	type of forwarding address	IN
class	class of information	IN
subaddr	subaddress	IN
satype	type of subaddress	IN
time	wait time before forwarding	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The sAT_PlusCCFC function controls the Call Forwarding supplementary service. The *reason* value distinguishes between the different types of forwarding services. Whereas the value of *mode* controls the action that should be performed. In case of registration the values of *number*, *type*, *subaddr*, *satype* and *time* define the forwarding address and behaviour. With the value of *class*, the call forwarding supplementary service operation could be addressed to the different classes of bearer services (voice, fax and data)

Related Functions:

qAT_PlusCCFC ()
 rAT_PlusCCFC ()
 rAT_PlusCME ()
 rAT_OK ()

6.3.1.7 sAT_PlusCCWA () - Call Waiting

Command Reference:

GSM 07.07 (7.11)

Function Definition:

T_ACI_RETURN sAT_PlusCCWA (T_ACI_CMD_SRC srcId, T_ACI_CCWA_MOD mode, T_ACI_CLASS class);

Parameters:

name	comment	
srcId	command source identifier	IN
mode	call waiting mode	IN
class	class of information	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The sAT_PlusCCWA function controls the Call Waiting supplementary service. The value of *mode* controls the action that should be performed. With the value of *class*, the Call Waiting supplementary service operation could be addressed to the different classes of bearer services (voice, fax and data)

Related Functions:

qAT_PlusCCWA ()
rAT_PlusCCWA ()
rAT_PlusCME ()
rAT_OK ()

6.3.1.8 sAT_PlusCHLD () - Call Hold and Multiparty

Command Reference:

GSM 07.07 (7.12)

Function Definition:

T_ACI_RETURN sAT_PlusCHLD (T_ACI_CMD_SRC srcId, T_ACI_CHLD_MOD mode, CHAR* call);

Parameters:

name	comment	
srcId	command source identifier	IN
mode	mode of operation	IN
call	call index	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The sAT_PlusCHLD function allows the current calls of the ME to be put on hold, recovered, released, added to conversation, and transferred similarly as defined in GSM 02.30. The *mode* value defines the action that should be performed, whereas the *index* value represents a specific call.

Related Functions:

rAT_PlusCME ()

rAT_OK ()

6.3.1.9 sAT_PlusCTFR () - Call Transfer (not implemented yet)**Command Reference:**

GSM 07.07 (7.13)

Function Definition:

```
T_ACI_RETURN sAT_PlusCTFR (T_ACI_CMD_SRC srcId, CHAR* number, T_ACI_TOA* type, CHAR* subaddr,
                           T_ACI_TOS* satype);
```

Parameters:

name	comment	
srcId	command source identifier	IN
number	forwarding address	IN
type	type of forwarding address	IN
subaddr	forwarding subaddress	IN
satype	type of forwarding subaddress	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The sAT_PlusCTFR function causes the incoming alerting call to be forwarded to a specified number. This is based on the GSM supplementary service Call Deflection GSM 02.72. Call Transfer is only applicable to teleservice 11. The buffer provided by *number* stores the forwarding phone number and the *type* value represents the respective type of address, whereas the buffer provided by *subaddr* stores the forwarding subaddress and the *satype* value represents the respective type of subaddress.

Related Functions:

rAT_OK ()

rAT_PlusCME ()

6.3.1.10 sAT_PlusCUSD () – Unstructured Supplementary Service Data

Command Reference:

GSM 07.07 (7.14)

Function Definition:

```
T_ACI_RETURN sAT_PlusCUSD (T_ACI_CMD_SRC srcId, T_ACI_USSD_DATA *str, SHORT dcs);
```

Parameters:

name	comment	
srcId	command source identifier	IN
str	USSD string	IN
dcs	data coding scheme	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Supplementary Services

Description:

The sAT_PlusCUSD function sends the USSD string in the location provided by the value *str* to the network, the coding of the USSD is given by the value *dcs*. When a USSD string is given, a mobile initiated USSD or a response USSD to a network initiated operation is sent to the network. In case of successful mobile initiated operation, the response USSD from the network is indicated to the application using the function rAT_PlusUSSD.

Related Functions:

rAT_PlusCUSD ()

rAT_OK ()

rAT_PlusCME ()

6.3.1.11 sAT_PlusCPOL() – Preferred Operator List**Command Reference:**

GSM 07.07 (7.18)

Function Definition:

```
T_ACI_RETURN sAT_PlusCPOL (T_ACI_CMD_SRC srcId, SHORT index,
                           T_ACI_CPOL_FRMT format, CHAR* oper);
```

Parameters:

name	comment	
srcId	command source identifier	IN
index	index to SIM preferred list of networks	IN
format	format of oper parameter	IN
oper	network operator	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Network

Description::

This function is used to write or delete an entry in the SIM list of preferred operators at the position given by the value of *index*. In the location *oper* the network operator is specified, the value *format* specifies the format in which the parameter *oper* is given. If *index* is given but *oper* is left out, the respective entry is deleted.

Related Functions:

```
qAT_PlusCPOL ()
rAT_PlusCPOL ()
tAT_PlusCPOL
rAT_OK ()
rAT_PlusCME ()
```

6.3.2 Query Functions

6.3.2.1 qAT_PlusCNUM () - Subscriber Number

Command Reference:

GSM 07.07 (7.1)

Function Definition:

T_ACI_RETURN qAT_PlusCNUM (T_ACI_CMD_SRC srcId, T_ACI_CNUM_MOD mode);

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
mode	---	distinguish between NewRead/NextRead	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Network

Description:

The qAT_PlusCNUM is used to read the list of MSISDNs related to the subscriber, which are located on the SIM. The value of *mode* defines if the first entry should be read or the next following entry. If the function completes successfully, no more entries are available. Otherwise the requested number is returned using the function rAT_PlusCNUM.

Related Functions:

rAT_PlusCNUM ()

rAT_PlusCME ()

rAT_OK()

6.3.2.2 qAT_PlusCREG () - Network Registration Info

Command Reference:

GSM 07.07 (7.2)

Function Definition:

```
T_ACI_RETURN qAT_PlusCREG (T_ACI_CMD_SRC srcId, T_ACI_CREG_STAT* stat);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
stat	---	registration status	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Network

Description:

The qAT_PlusCREG function returns the current network registration status. The return value is stored in the location provided by *stat* if the function returns successfully.

Related Functions:

6.3.2.3 qAT_PlusCOPS () - Operator Selection

Command Reference:

GSM 07.07 (7.3)

Function Definition:

```
T_ACI_RETURN qAT_PlusCOPS (T_ACI_CMD_SRC srcId, T_ACI_COPS_MOD* mode,  
                           T_ACI_COPS_FRMT* format, CHAR* oper);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
mode	---	network registration mode	OUT
format	---	operator format	OUT
oper	MAX_ALPHA_OPER_LEN	selected operator	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Network

Description:

The qAT_PlusCOPS function returns the current setting of the parameters defining the operator selection, including registration mode and operator format, and the currently selected operator. The return values are stored in the respective locations provided by *mode*, *format* and *oper* if the function returns successfully.

Related Functions:

rAT_PlusCOPS ()

sAT_PlusCOPS ()

tAT_PlusCOPS ()

6.3.2.4 qAT_PlusCLCK () - Facility Lock**Command Reference:**

GSM 07.07 (7.4)

Function Definition:

```
T_ACI_RETURN qAT_PlusCLCK (T_ACI_CMD_SRC srcId, T_ACI_CLCK_FAC fac,
                           T_ACI_CLASS class, T_ACI_CLSSTAT* clsStat);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
fac	---	requested facility	IN
class	---	service class	IN
clsStat	---	facility status	OUT

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	successfully completed
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The qAT_PlusCLCK function returns the current setting for the requested facility *fac*. The value of *class* is used to define the bearer services for the query (voice, fax and data). The returned setting is stored in the location provided by *clsStat* if the function returns successfully. If the processing of the request can not be completed immediately, the function returns with the value AT_EXCT, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCLCK.

Related Functions:

```
rAT_PlusCLCK ()
sAT_PlusCLCK ()
rAT_PlusCME ()
rAT_OK()
```

6.3.2.5 qAT_PlusCLIP () - Calling Line Identification Presentation

Command Reference:

GSM 07.07 (7.6)

Function Definition:

```
T_ACI_RETURN qAT_PlusCLIP (T_ACI_CMD_SRC srcId, T_ACI_CLIP_STAT* stat);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
stat	---	CLIP network service status	OUT

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	completed successfully
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The qAT_PlusCLIP function returns the current setting of the CLIP service status in the network. The returned value is stored in the location provided by *stat* if the function returns successfully. If the processing of the request can not be completed immediately, the function returns with the value AT_EXCT, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCLIP.

Related Functions:

rAT_PlusCLIP ()

rAT_PlusCME ()

rAT_OK()

6.3.2.6 qAT_PlusCLIR () - Calling Line Identification Restriction

Command Reference:

GSM 07.07 (7.7)

Function Definition:

```
T_ACI_RETURN qAT_PlusCLIR (T_ACI_CMD_SRC srcId, T_ACI_CLIR_MOD* mode,  
                           T_ACI_CLIR_STAT* stat);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
mode	---	mode for CLIR	OUT
stat	---	CLIR network service status	OUT

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	successfully completed
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The qAT_PlusCLIR function returns the current setting of the CLIR mode and the CLIR service status in the network. The values returned are stored in the locations provided by *mode* and *stat* if the function returns successfully. If the processing of the request can not be completed immediately, the function returns with the value AT_EXCT, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCLIR.

Related Functions:

rAT_PlusCLIR ()

sAT_PlusCLIR ()

rAT_PlusCME ()

rAT_OK()

6.3.2.7 qAT_PlusCOLP () - Connected Line Identification Presentation

Command Reference:

GSM 07.07 (7.8)

Function Definition:

```
T_ACI_RETURN qAT_PlusCOLP (T_ACI_CMD_SRC srcId, T_ACI_COLP_STAT* stat);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
stat	---	COLP network service status	OUT

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	successfully completed
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The qAT_PlusCOLP function returns the current setting of the COLP service status in the network. The returned value is stored in the location provided by *stat* if the function returns successfully. If the processing of the request can not be completed immediately, the function returns with the value AT_EXCT, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCOLP.

Related Functions:

rAT_PlusCOLP ()

rAT_PlusCME ()

rAT_OK()

6.3.2.8 qAT_PlusCCUG () - Closed User Group

Command Reference:

GSM 07.07 (7.9)

Function Definition:

```
T_ACI_RETURN qAT_PlusCCUG (T_ACI_CMD_SRC srcId, T_ACI_CCUG_MOD* mode,  
                           T_ACI_CCUG_IDX* index, T_ACI_CCUG_INFO* info);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
mode	---	CUG temporary mode	OUT
index	---	CUG index	OUT
info	---	information	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Supplementary Services

Description:

The qAT_PlusCCUG function returns the current setting of the Closed User Group parameters, including the CUG temporary mode, the CUG index and the CUG information. The returned values are stored in the locations provided by *mode*, *index* and *info* if the function returns successfully.

Related Functions:

sAT_PlusCCUG ()

6.3.2.9 qAT_PlusCCFC () - Call Forwarding Number and Conditions

Command Reference:

GSM 07.07 (7.10)

Function Definition:

```
T_ACI_RETURN qAT_PlusCCFC (T_ACI_CMD_SRC srcId, T_ACI_CCFC_RSN reason,  
                           T_ACI_CLASS class);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
reason	---	requested call forwarding condition	IN
class	---	service class	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The qAT_PlusCCFC function returns the current setting for the requested Call Forwarding condition given by the value of *reason*. The value of *class* is used to define the different bearer services for the query (voice, fax and data). The function returns with the value AT_EXCT in case of a valid query, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCCFC.

Related Functions:

rAT_PlusCCFC ()
sAT_PlusCCFC ()
rAT_PlusCME ()
rAT_OK()

6.3.2.10 qAT_PlusCCWA () - Call Waiting

Command Reference:

GSM 07.07 (7.11)

Function Definition:

```
T_ACI_RETURN qAT_PlusCCWA (T_ACI_CMD_SRC srcId, T_ACI_CLASS class);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
class	---	service class	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Supplementary Services

Description:

The qAT_PlusCCWA function returns the current setting for the Call Waiting supplementary service. The value of *class* is used to define the different bearer services for the query (voice, fax and data). The function returns with the value AT_EXCT in case of a valid query, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCCWA.

Related Functions:

rAT_PlusCCWA ()

sAT_PlusCCWA ()

rAT_PlusCME ()

rAT_OK()

6.3.2.11 qAT_PlusCAOC () – Advice of Charge

Command Reference:

GSM 07.07 (7.15)

Function Definition:

```
T_ACI_RETURN qAT_PlusCAOC (T_ACI_CMD_SRC srcId, LONG* ccm);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
ccm	---	current call meter	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Supplementary Services

Description:

The qAT_PlusCAOC function returns the value of the current call meter. The returned value is stored in the location provided by the value *ccm*.

Related Functions:

6.3.2.12 qAT_PlusCLCC () – List Current Calls

Command Reference:

GSM 07.07 (7.17)

Function Definition:

```
T_ACI_RETURN qAT_PlusCLCC (T_ACI_CMD_SRC srcId, T_ACI_CLCC_CALDESC* callLst);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
callLst	MAX_CALL_NR	list of current calls	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Call Control

Description:

The qAT_PlusCLCC function returns the status of the current calls of the ME. The returned status for the calls are stored in the location provided by the value *callLst*.

Related Functions:

6.3.2.13 qAT_PlusCPOL () – Preferred Operator List**Command Reference:**

GSM 07.07 (7.18)

Function Definition:

```
T_ACI_RETURN qAT_PlusCPOL (T_ACI_CMD_SRC srcId, SHORT startIdx,
                           SHORT *lastIdx, T_ACI_CPOL_OPDESC *operLst);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
startIdx	---	first operator to store	IN
lastIdx	---	last operator stored	IN/OUT
operLst	MAX_OPER	operator list	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Network

Description:

The function qAT_PlusCPOL () queries the SIM for the preferred operator list. The array *operLst* is filled with the respective operator data. The value *startIdx* gives the index of the first operator to store in *operLst*. The last index returned in *operLst* is returned in the location given by *lastIdx*. The maximum number of entries stored by one call is MAX_OPER. If the number of preferred operators exceeds the list size, successive function calls to qAT_PlusCPOL are necessary.

The function returns with the value AT_EXCT in case that the query can not be completed immediately and is still being executed. The final result will then be passed to the caller via the respective call-back function rAT_PlusCPOL.

Related Functions:

```
rAT_PlusCPOL ()
sAT_PlusCPOL ()
tAT_PlusCPOL ()
rAT_PlusCME ()
rAT_OK()
```

6.3.2.14 qAT_PlusCOPN () – Read Operator Names**Command Reference:**

GSM 07.07 (7.19)

Function Definition:

```
T_ACI_RETURN qAT_PlusCOPN (T_ACI_CMD_SRC srcId, T_ACI_COPN_LID lstId,
                           SHORT startIdx, SHORT *lastIdx, T_ACI_COPN_OPDESC *oprLstBuf);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
lstId	---	list identifier	IN
startIdx	---	start index to read from	IN
lastIdx	---	index of last copied name	OUT
oprLstBuf	MAX_OPER	buffer for operator names	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Network

Description:

The function qAT_plusCOPN returns the list of operator names stored in the ME fixed and configuration memory. The value *lstId* determines the ME memory list to be read from. The value *startIdx* determines the start index into this memory. The index of the last copied name is stored in the buffer location provided by the value *lastIdx*. The value *oprLstBuf* provides the buffer location where the operators are returned in alphanumerical and numerical format. The maximum number of entries stored by one call is MAX_OPER. . If the number of operator names exceeds the list size, successive function calls to qAT_PlusCOPN are necessary.

Related Functions:

6.3.3 Test Functions

6.3.3.1 tAT_PlusCOPS () - Operator Selection

Command Reference:

GSM 07.07 (7.3)

Function Definition:

```
T_ACI_RETURN tAT_PlusCOPS (T_ACI_CMD_SRC srcId, SHORT startIdx, SHORT* lastIdx,
                           T_ACI_COPS_OPDESC* operLst);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
startIdx	---	index of the next requested operator	IN
lastIdx	---	index of the last passed operator	OUT
operLst	MAX_OPER	list of operators	OUT

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	successfully completed
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Network

Description:

The tAT_PlusCOPS function returns a list of all operators present in the network. The result is stored in the location provided by *operLst* if the function returns successfully. The value defined by MAX_OPER (see constant definitions) limits the number of entries which are copied into the buffer at the location *operLst*. If the number of operators exceeds the list size, successive function calls to tAT_PlusCOPS are necessary.

If the processing of the request can not be completed immediately, the function returns with the value AT_EXCT, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCOPS.

When calling the tAT_PlusCOPS function, *startIdx* provides the index of the next requested entry in the operator list. *startIdx* must be greater than or equal 0. If the function returns successfully, *lastIdx* points to the index of the last copied list entry.

Related Functions:

qAT_PlusCOPS ()
 rAT_PlusCOPS ()
 sAT_PlusCOPS ()
 rAT_PlusCME ()
 rAT_OK()

6.3.3.2 tAT_PlusCPOL () – Preferred Operator List**Command Reference:**

GSM 07.07 (7.18)

Function Definition:

T_ACI_RETURN tAT_PlusCPOL (T_ACI_CMD_SRC srcId, SHORT* lastIdx, SHORT usdNtry);

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
lastIdx	---	number of supported entries	OUT
usdNtry	---	number of used entries	OUT

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	successfully completed
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Network

Description:

This function returns the number of supported entries and used entries of the SIM preferred list of networks. If the processing of the request can not be processed immediately, the function returns with the value AT_EXCT, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCPOL.

Related Functions:

qAT_PlusCPOL()
 rAT_PlusCPOL()
 sAT_PLusCPOL ()
 rAT_PlusCME ()
 rAT_OK()

6.3.4 Call-back Functions

6.3.4.1 rAT_PlusCNUM () - Subscriber Number

Command Reference:

GSM 07.07 (7.1)

Function Definition:

```
void rAT_PlusCNUM (T_ACI_CNUM_MSISDN* msisdn, UBYTE num);
```

Parameters:

name	comment	
msisdn	list of subscriber related MSISDNs	IN
num	maximum number of MSISDN	IN

Return:

none

Function Group:

Network

Description:

This function is used to return the requested subscriber related MSISDNs if the request using the qAT_PlusCNUM was not completed immediately. A reference to the requested number is passed via *msisdnLst*. The maximum number of stored MSISDN is returned by the value of *num*.

Related Functions:

qAT_PlusCNUM ()

6.3.4.2 rAT_PlusCOPS () - Operator Selection

Command Reference:

GSM 07.07 (7.3)

Function Definition:

```
void rAT_PlusCOPS (SHORT lastIdx, T_ACI_COPS_OPDESC* operLst);
```

Parameters:

name	comment	
lastIdx	last entry index	IN
operLst	list of operators	IN

Return:

none

Function Group:

Network

Description:

This function is used to return the requested list of operators if the request using the tAT_PlusCOPS was not completed immediately. A reference to the requested list is passed via *operLst*. The *lastIdx* value indicates the index of the last *operLst* entry in reference to the entire list of operators found. If the length of *operLst* is not sufficient enough to hold the complete list, a new request starting with *lastIdx+1* can be made to get the next part of the list. (See MAX_OPER in the section of constant definitions.)

Related Functions:

qAT_PlusCOPS ()

sAT_PlusCOPS ()

tAT_PlusCOPS ()

6.3.4.3 rAT_PlusCREG () - Network Registration

Command Reference:

GSM 07.07 (7.2)

Function Definition:

```
void rAT_PlusCREG (T_ACI_CREG_STAT status);
```

Parameters:

name	comment	
status	network registration status	IN

Return:

none

Function Group:

Results

Description:

The rAT_PlusCREG function is used to report when there is a change in the network registration status. The value of *stat* indicates the new network status.

Related Functions:

qAT_PlusCREG ()

6.3.4.4 rAT_PlusCLCK () - Facility Lock

Command Reference:

GSM 07.07 (7.4)

Function Definition:

```
void rAT_PlusCLCK (T_ACI_CLSSTAT* clsStatLst);
```

Parameters:

name	comment	
clsStatLst	facility status list	IN

Return:

none

Function Group:

Supplementary Services

Description:

The rAT_PlusCLCK function is used to return the setting for the requested facility if the request using the qAT_PlusCLCK was not completed immediately. A reference to the requested setting is passed via *clsStatLst*.

Related Functions:

qAT_PlusCLCK ()

sAT_PlusCLCK ()

6.3.4.5 rAT_PlusCLIP () - Calling Line Identification Presentation**Command Reference:**

GSM 07.07 (7.6)

Function Definition:

```
void rAT_PlusCLIP (T_ACI_CLIP_STAT stat, CHAR* number, T_ACI_TOA* type, CHAR* subaddr,
                  T_ACI_TOS* satype, CHAR* alpha);
```

Parameters:

name	comment	
stat	CLIP network service status	IN
number	number of calling party	IN
type	type of address of number of calling party	IN
subaddr	subaddress of calling party	IN
satype	type of subaddress of calling party	IN
alpha	alphanumeric identification of calling party	IN

Return:

none

Function Group:

Supplementary Services

Description:

The rAT_PlusCLIP function is used in two ways. The first is to present the calling line identification for an incoming call. In this case, depending on the information available from this particular call, *number*, *type*, *subaddr*, *satype* and *alpha* can contain the presentation information.

Secondly, the rAT_PlusCLIP function is used as a return for the requested setting of the CLIP parameters if the request using the qAT_PlusCLIP was not completed immediately. The *stat* value indicate the current setting for CLIP supplementary service in the network.

Related Functions:

qAT_PlusCLIP ()

6.3.4.6 rAT_PlusCLIR () - Calling Line Identification Restriction

Command Reference:

GSM 07.07 (7.7)

Function Definition:

```
void rAT_PlusCLIR (T_ACI_CLIR_MOD mode, T_ACI_CLIR_STAT stat);
```

Parameters:

name	comment	
mode	mode for CLIR	IN
stat	CLIR network service status	IN

Return:

none

Function Group:

Supplementary Services

Description:

The rAT_PlusCLIR function is used to return the requested CLIR parameters if the request using the qAT_PlusCLIR was not completed immediately. The requested CLIR parameters are passed via *mode* and *stat*.

Related Functions:

qAT_PlusCLIR ()

sAT_PlusCLIR ()

6.3.4.7 rAT_PlusCOLP () - Connected Line Identification Presentation

Command Reference:

GSM 07.07 (7.8)

Function Definition:

```
void rAT_PlusCOLP (T_ACI_COLP_STAT stat, CHAR* number, T_ACI_TOA* type, CHAR* subaddr,  
                  T_ACI_TOS* satype, CHAR* alpha);
```

Parameters:

name	comment	
stat	COLP network service status	IN
number	number of connected line	IN
type	type of address of connected line	IN
subaddr	subaddress of connected line	IN
satype	type of subaddress of connected line	IN
alpha	alphanumeric identification of connected line	IN

Return:

none

Function Group:

Supplementary Services

Description:

The rAT_PlusCOLP function is used in two ways. The first is to present the connected line identification for a mobile originated call. In this case, dependent on the information available for this particular call, *number*, *type*, *subaddr*, *satype* and *alpha* can contain the presentation information.

Secondly, the rAT_PlusCLOP function is used as a return for the requested setting of the COLP parameters if the request using the qAT_PlusCOLP was not completed immediately. The *stat* value indicates the current setting for COLP supplementary service in the network.

Related Functions:

qAT_PlusCOLP ()

6.3.4.8 rAT_PlusCCFC () - Call Forwarding Number and Conditions

Command Reference:

GSM 07.07 (7.10)

Function Definition:

```
void rAT_PlusCCFC (T_ACI_CCFC_SET* setting);
```

Parameters:

name	comment	
setting	call forwarding setting	IN

Return:

none

Function Group:

Supplementary Services

Description:

The rAT_PlusCCFC function is used to return the setting for the requested call forwarding condition if the request using the qAT_PlusCCFC was not completed immediately. A reference to the requested setting is passed via *setting*.

Related Functions:

qAT_PlusCCFC ()

sAT_PlusCCFC ()

6.3.4.9 rAT_PlusCCWA () - Call Waiting

Command Reference:

GSM 07.07 (7.11)

Function Definition:

```
void rAT_PlusCCWA (T_ACI_CLSSTAT* clsStat, CHAR* number, T_ACI_TOA* type,  
                  T_ACI_CLASS class, CHAR* alpha);
```

Parameters:

name	comment	
clsStat	call waiting status	IN
number	number of waiting call	IN
type	type of address of waiting call	IN
class	class of waiting call	IN
alpha	alphanumeric identification of waiting call	IN

Return:

none

Function Group:

Supplementary Services, Result

Description:

The rAT_PlusCCWA function is used in two ways. The first is to present an identification for a waiting call. In this case, dependent on the information available for this call, *number*, *type*, *class* and *alpha* can contain the presentation information.

Secondly, the rAT_PlusCCWA function is used as a return for the requested setting for the call waiting presentation mode, if the request using the qAT_PlusCCWA was not completed immediately. A reference to a status *clsStat* indicates the current setting for CCWA supplementary service in the network.

Related Functions:

qAT_PlusCCWA ()

sAT_PlusCCWA ()

6.3.4.10 rAT_PlusCUSD () – Unstructured Supplementary Service Data**Command Reference:**

GSM 07.07 (7.14)

Function Definition:

```
void rAT_PlusCUSD (T_ACI_CUSD_MOD m, T_ACI_USSD_DATA* ussd, SHORT dcs);
```

Parameters:

name	comment	
m	USSD mode	IN
ussd	USSD data	IN
dcs	data coding scheme	IN

Return:

none

Function Group:

Supplementary Services, Result

Description:

The rAT_PlusCUSD function is used to notify the application about an incoming USSD message or an incoming USSD response to a previous USSD request using the function sAT_PlusCUSD.

Related Functions:

sAT_PlusCUSD ()

6.3.4.11 rAT_PlusCCCM () – Change Current Call Meter

Command Reference:

GSM 07.07 (7.15)

Function Definition:

```
void rAT_PlusCCCM (LONG *ccm);
```

Parameters:

name	comment	
ccm	current call meter	IN

Return:

none

Function Group:

Supplementary Services, Result

Description:

The rAT_PlusCCCM function is used to notify the application about a change of the current call meter. The new current call meter value is located at the location given by *ccm*.

Related Functions:

sAT_PlusCAOC ()

qAT_PlusCAOC

6.3.4.12 rAT_PlusCSSI () – Supplementary Service Notification

Command Reference:

GSM 07.07 (7.16)

Function Definition:

void rAT_PlusCSSI (T_ACI_CSSSI_CODE code, SHORT index);

Parameters:

name	comment	
code	type of supplementary service message	IN
index	closed user group index	IN

Return:

none

Function Group:

Supplementary Services, Result

Description:

The rAT_PLusCSSI function informs the application about the reception of supplementary service notifications after a mobile originated call. The kind of information is indicated by the value of *code*, the value of *index* is given in case that the call is a CUG call.

Related Functions:

rAT_PlusCSSU ()

6.3.4.13 rAT_PlusCSSU () – Supplementary Service Notification

Command Reference:

GSM 07.07 (7.16)

Function Definition:

```
void rAT_PlusCSSU (T_ACI_CSSI_CODE code, SHORT index, CHAR* number,  
                  T_ACI_TOA* type, CHAR*subaddr, T_ACI_TOS* satype);
```

Parameters:

name	comment	
code	type of supplementary service message	IN
index	closed user group index	IN
number	phone number	IN
type	type of address	IN
subaddr	subaddress	IN
satype	type of subaddress	IN

Return:

none

Function Group:

Supplementary Services, Result

Description:

The rAT_PlusCSSU function informs the application about the reception of supplementary service notifications during a mobile terminated call setup or during a call. The kind of information is indicated by the value of *code*, the value of *index* is given in case that the call is a CUG call. In case of the completion of an explicit call transfer operation the values of *number*, *type*, *subaddr* and *satype* may contain information about the remote party.

Related Functions:

rAT_PlusCSSI ()

6.3.4.14 rAT_PlusCPOL () – Preferred Operator List**Command Reference:**

GSM 07.07 (7.18)

Function Definition:

```
void rAT_PlusCPOL (SHORT startIdx, SHORT lastIdx, T_ACI_CPOL_OPDESC* operLst, SHORT usdNtry);
```

Parameters:

name	buffer size	comment	
startIdx	---	start index	IN
lastIdx	---	last index	IN
operLst	MAX_OPER	operator list	IN
usdNtry	---	number of used entries	IN

Return:

none

Function Group:

Network

Description:

The rAT_PlusCPOL function is used in two ways. The first is to inform the application about the preferred operator list. The value of *startIdx* indicates the first requested entry of the list and the value *lastIdx* represents the number of the last valid operator in the array *operLst*. The operator data is given in the location provided by *operLst*. The maximum amount of valid entries in *operLst* is MAX_OPER.

Secondly, the rAT_PlusCPOL function is used as a return for the requested status of the preferred operator list if the request using the tAT_PlusCPOL was not completed immediately. The *lastIdx* value indicates the maximum number of entries and *usdNtry* indicates the number of used entries.

Related Functions:

qAT_PlusCPOL ()

sAT_PlusCPOL ()

tAT_PlusCPOL ()

6.4 ME Control and Status Functions

6.4.1 Set Functions

6.4.1.1 sAT_PlusCFUN () - Set Phone Functionality

Command Reference:

GSM 07.07 (8.2)

Function Definition:

T_ACI_RETURN sAT_PlusCFUN (T_ACI_CMD_SRC srcId, T_ACI_CFUN_FUN fun, T_ACI_CFUN_RST rst);

Parameters:

name	comment	
srcId	command source identifier	IN
fun	phone functionality	IN
rst	reset parameter	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Phone Control

Description:

The sAT_PlusCFUN function selects the level of functionality in the ME. The *fun* value represents the state of functionality and the *rst* value - whether or not the ME is reset.

Related Functions:

qAT_PlusCFUN ()
rAT_PlusCME ()
rAT_OK ()

6.4.1.2 sAT_PlusCPIN () - Enter PIN

Command Reference:

GSM 07.07 (8.3)

Function Definition:

```
T_ACI_RETURN sAT_PlusCPIN (T_ACI_CMD_SRC srcId, CHAR* pin, CHAR* newpin);
```

Parameters:

name	comment	
srcId	command source identifier	IN
pin	PIN (password)	IN
newpin	new PIN (password)	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Phone Control

Description:

The sAT_PlusCPIN function sends a password to the ME, which is necessary before it can be operated (SIM PIN, SIM PUK, etc.). The PIN/PUK which is currently requested by ME could be queried using the qAT_PlusCPIN function. If the PIN required is SIM PUK, the second pin is required. This second PIN, *newpin*, is used to replace the old PIN in the SIM.

Related Functions:

qAT_PlusCPIN ()

rAT_PlusCME ()

rAT_OK()

6.4.1.3 sAT_PlusCPBS () - Select Phonebook Memory Storage

Command Reference:

GSM 07.07 (8.11)

Function Definition:

```
T_ACI_RETURN sAT_PlusCPBS (T_ACI_CMD_SRC srcId, T_ACI_PB_STOR mem);
```

Parameters:

name	comment	
srcId	command source identifier	IN
mem	phonebook memory storage	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Phonebook Control

Description:

The sAT_PlusCPBS function selects the phonebook memory storage, which is used by other phonebook functions as a preselection.

Related Functions:

qAT_PlusCPBS ()

6.4.1.4 sAT_PlusCPBR () - Read Phonebook Entries**Command Reference:**

GSM 07.07 (8.12)

Function Definition:

```
T_ACI_RETURN sAT_PlusCPBR (T_ACI_CMD_SRC srcId, SHORT startIdx, SHORT stopIdx,
                           SHORT* lastIdx, T_ACI_PB_ENTR* pbLst);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
startIdx	---	index of first location to be read	IN
stopIdx	---	index of last location to be read	IN
lastIdx	---	index of last location already read	OUT
pbLst	MAX_PB_ENTR	list of phonebook entries	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Phonebook Control

Description:

The sAT_PlusCPBR function returns the phone book entries in the location number range specified by *startIdx* to *stopIdx* from the currently selected phonebook. If *stopIdx* is declared as not present, only the entry at location *startIdx* is returned. The returned entries are stored in the location provided by *pbLst* if the function returns successfully. The maximum number of entries stored by one call is MAX_PB_ENTR. If the number of requested phonebook entries exceeds the list size, successive function calls to sAT_PlusCPBR are necessary.

Related Functions:

tAT_PlusCPBR ()

6.4.1.5 sAT_PlusCPBF () - Find Phonebook Entries**Command Reference:**

GSM 07.07 (8.13)

Function Definition:

```
T_ACI_RETURN sAT_PlusCPBF (T_ACI_CMD_SRC srcId, CHAR* findtext, T_ACI_CPBF_MOD mode,
                           UBYTE* found, T_ACI_PB_ENTR* pbLst);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
findtext	---	search string	IN
mode	---	search mode	IN
found	---	number of entries found	OUT
pbLst	MAX_PB_ENTR	list of entries found	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Phonebook Control

Description:

The sAT_PlusCPBF function is used to search for an entry in the currently selected phonebook. The phonebook is searched for entries that matches the alphanumeric search string *findtext*. The search result is stored in the location provided by *pbLst* if the function returns successfully. The value of *found* gives the total number of found entries. This value can be used to decide whether more entries are retrievable by further calls to function sAT_PlusCPBF. Via *mode* a new search could be started or the next available found entries could be retrieved.

Related Functions:

tAT_PlusCPBF ()

6.4.1.6 sAT_PlusCPBW () - Write Phonebook Entry**Command Reference:**

GSM 07.07 (8.14)

Function Definition:

```
T_ACI_RETURN sAT_PlusCPBW (T_ACI_CMD_SRC srcId, SHORT index, CHAR* number,
                             T_ACI_TOA* type, CHAR* text T_ACI_VP_ABS* dateTime);
```

Parameters:

name	comment	
srcId	command source identifier	IN
index	value in the range of location numbers of phonebook memory	IN
number	phone number	IN
type	type of address of phone number	IN
text	text string associated with phone number	IN
dateTime	date, time and timezone	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Phonebook Control

Description:

The sAT_PlusCPBW function writes a phonebook entry in the current phonebook memory storage selected with sAT_PlusCPBS. The entry fields written are the phone number and text associated with that number. If those fields are omitted and the *index* value is given, the phonebook entry is deleted. If the *index* is left out, but the phone number is provided, the entry is written to the first free location in the phonebook. If *dateTime* is NULL, date and time of the phone book entry will be invalidated.

Related Functions:

tAT_PlusCPBW ()

6.4.1.7 sAT_PlusCACM () – Accumulated Call Meter**Command Reference:**

GSM 07.07 (8.25)

Function Definition:

```
T_ACI_RETURN sAT_PlusCACM (T_ACI_CMD_SRC srcId, CHAR* passwd);
```

Parameters:

name	comment	
srcId	command source identifier	IN
passwd	password	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed

Function Group:

Advice of Charge

Description:

The sAT_PlusCACM function resets the accumulated call meter in the SIM. Usually the SIM PIN2 is required in the location provided by the value *passwd* for this function.

Related Functions:

qAT_PlusCACM ()
rAT_PlusCME ()
rAT_OK()

6.4.1.8 sAT_PlusCamm () – Accumulated Call Meter Maximum**Command Reference:**

GSM 07.07 (8.26)

Function Definition:

```
T_ACI_RETURN sAT_PlusCamm (T_ACI_CMD_SRC srcId, LONG acmmax, CHAR* passwd);
```

Parameters:

name	comment	
srcId	command source identifier	IN
acmmax	maximum allowed accumulated call meter	IN
passwd	password	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed

Function Group:

Advice of Charge

Description:

The sAT_PlusCamm function stores the value *acmmax* to the accumulated call meter maximum ACMmax in the SIM. Usually the SIM PIN2 is required in the location provided by the value *passwd* for this function.

Related Functions:

qAT_PlusCamm ()

rAT_PlusCME ()

rAT_OK()

6.4.1.9 sAT_PlusCPUC () – Price per Unit and Currency Table**Command Reference:**

GSM 07.07 (8.27)

Function Definition:

```
T_ACI_RETURN sAT_PlusCPUC (T_ACI_CMD_SRC srcId, CHAR*cur, CHAR* ppu, CHAR* passwd);
```

Parameters:

name	comment	
srcId	command source identifier	IN
cur	currency	IN
ppu	price per unit	IN
passwd	password	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed

Function Group:

Advice of Charge

Description:

The sAT_PlusCPUC function sets the price per unit and currency table. The values *cur* and *ppu* are stored in the SIM. Usually the SIM PIN2 is required in the location provided by the value *passwd* for this function.

Related Functions:

qAT_PlusCMM ()

rAT_PlusCME ()

rAT_OK()

6.4.1.10 sAT_PlusCRSM() – Restricted SIM Access**Command Reference:**

GSM 07.07 (8.18)

Function Definition:

```
T_ACI_RETURN sAT_PlusCRSM (T_ACI_CMD_SRC srcId, T_ACI_CRSM_CMD cmd, SHORT fileId,
                           SHORT p1, SHORT p2, SHORT p3, SHORT dataLen, UBYTE * data );
```

Parameters:

name	comment	
srcId	command source identifier	IN
cmd	SIM access command	IN
fileId	file identifier	IN
p1	command parameter p1	IN
p2	command parameter p2	IN
p3	command parameter p3	IN
dataLen	length of data in bytes	IN
data	pointer to data to be written to SIM	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

SIM access

Description:

This function is used to access SIM data in a restricted way. The parameter *cmd* defines the type of access to the SIM that should be performed. The file that is accessed is defined by the parameter *fileId*. According to GSM 11.11 the parameter *p1*, *p2* and *p3* provide supplemental information for the SIM command. If data should be updated on the SIM, a pointer to a string of bytes *data* must be provided. The number of data bytes has to be indicated using the parameter *dataLen*.

Related Functions:

rAT_PlusCRSM ()

6.4.2 Query Functions

6.4.2.1 qAT_PlusCPAS () - Phone Activity Status

Command Reference:

GSM 07.07 (8.1)

Function Definition:

T_ACI_RETURN qAT_PlusCPAS (T_ACI_CMD_SRC srcId, T_ACI_CPAS_PAS* pas);

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
pas	---	phone activity status	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

ME Control and Status Commands

Description:

The qAT_PlusCPAS function returns the current phone activity status. The returned value is stored in the location provided by *pas* if the function returns successfully.

Related Functions:

6.4.2.2 qAT_PlusCFUN () - Phone Functionality

Command Reference:

GSM 07.07 (8.2)

Function Definition:

```
T_ACI_RETURN qAT_PlusCFUN (T_ACI_CMD_SRC srcId, T_ACI_CFUN_FUN* fun);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
fun	---	phone functionality status	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Phone Control

Description:

The qAT_PlusCFUN function returns the current setting for the phone functionality level. The returned value is stored in the location provided by *fun* if the function returns successfully.

Related Functions:

sAT_PlusCFUN ()

6.4.2.3 qAT_PlusCPIN () - Enter PIN

Command Reference:

GSM 07.07 (8.3)

Function Definition:

T_ACI_RETURN qAT_PlusCPIN (T_ACI_CMD_SRC srcId, T_ACI_CPIN_RSLT* code);

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
code	---	PIN status code	OUT

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Phone Control

Description:

The qAT_PlusCPIN function returns the current status for the PIN or PUK entry. The returned status is stored in the location provided by *code* if the function returns successfully. If the processing of the request can not be completed immediately, the function returns with the value AT_EXCT, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCPIN.

Related Functions:

sAT_PlusCPIN ()

rAT_PlusCPIN ()

rAT_PlusCME ()

rAT_OK()

6.4.2.4 qAT_PlusCPBS () - Select Phonebook Memory Storage

Command Reference:

GSM 07.07 (8.11)

Function Definition:

```
T_ACI_RETURN qAT_PlusCPBS (T_ACI_CMD_SRC srcId, T_ACI_PB_STOR* storage, UBYTE* used,
                           UBYTE* total);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
storage	---	selected phonebook memory storage	OUT
used	---	used entries	OUT
total	---	total entries	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Phonebook Control

Description:

The qAT_PlusCPBS function returns the currently selected phone book, including the number of used and total entries. The returned values are stored in the respective locations provided by *storage*, *used* and *total* if the function returns successfully.

Related Functions:

sAT_PlusCPBS ()

6.4.2.5 qAT_PlusCACM () – Accumulated Call Meter

Command Reference:

GSM 07.07 (8.25)

Function Definition:

```
T_ACI_RETURN qAT_PlusCACM (T_ACI_CMD_SRC srcId, LONG* acm);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
acm	---	accumulated call meter	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed

Function Group:

Advice of Charge

Description:

The qAT_PlusCACM function returns the accumulated call meter. The returned value is stored in the location provided by the value *acm*.

Related Functions:

sAT_PlusCACM ()

6.4.2.6 qAT_PlusCamm () – Accumulated Call Meter Maximum**Command Reference:**

GSM 07.07 (8.26)

Function Definition:

```
T_ACI_RETURN qAT_PlusCamm (T_ACI_CMD_SRC srcId, LONG* acmmax);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
acmmax	---	accumulated call meter maximum	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed

Function Group:

Advice of Charge

Description:

The qAT_PlusCamm function returns the accumulated call meter maximum. The returned value is stored in the location provided by the value *acmmax*.

Related Functions:

sAT_PlusCamm ()

6.4.2.7 qAT_PlusCPUC () – Price per Unit and Currency Table

Command Reference:

GSM 07.07 (8.27)

Function Definition:

```
T_ACI_RETURN qAT_PlusCPUC (T_ACI_CMD_SRC srcId, CHAR* cur, CHAR* ppu);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
cur	MAX_CUR_LEN	currency	OUT
ppu	MAX_NUM_LEN	price per unit	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Advice of Charge

Description:

The qAT_PlusCPUC function returns the price per unit and currency table information. The currency is stored in the location provided by the value *cur*, the price per unit is stored in the location provided by the value *ppu*.

Related Functions:

sAT_PlusCPUC ()

6.4.3 Test Functions

6.4.3.1 tAT_PlusCPBR () – Read Phonebook Entries

Command Reference:

GSM 07.07 (8.12)

Function Definition:

T_ACI_RETURN tAT_PlusCPBR (T_ACI_CMD_SRC srcId, SHORT* firstIdx, SHORT* lastIdx,
UBYTE* nlength, UBYTE* tlength)

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
firstIdx	---	first supported index	OUT
lastIdx	---	last supported index	OUT
nlength	---	maximum length of phone number	OUT
tlength	---	maximum length of associated text	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Phonebook Control

Description:

The tAT_PlusCPBR function returns the location range supported by the current storage, the maximum length of phone numbers and the maximum length of text fields associated with these numbers.

Related Functions:

sAT_PlusCPBR ()

6.4.3.2 tAT_PlusCPBF () – Find Phonebook Entries

Command Reference:

GSM 07.07 (8.13)

Function Definition:

T_ACI_RETURN tAT_PlusCPBF (T_ACI_CMD_SRC srcId, UBYTE* nlength, UBYTE* tlength)

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
nlength	---	maximum length of phone number	OUT
tlength	---	maximum length of associated text	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Phonebook Control

Description:

The tAT_PlusCPBF function returns the maximum length of phone numbers and the maximum length of text fields associated with these numbers.

Related Functions:

sAT_PlusCPBF ()

6.4.3.3 tAT_PlusCPBW () – Write Phonebook Entry**Command Reference:**

GSM 07.07 (8.14)

Function Definition:

T_ACI_RETURN tAT_PlusCPBW (T_ACI_CMD_SRC srcId, SHORT* firstIdx, SHORT* lastIdx,
 UBYTE* nlength, UBYTE* tlength)

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
firstIdx	---	first supported index	OUT
lastIdx	---	last supported index	OUT
nlength	---	maximum length of phone number	OUT
tlength	---	maximum length of associated text	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Phonebook Control

Description:

The tAT_PlusCPBW function returns the location range supported by the current storage, the maximum length of phone numbers and the maximum length of text fields associated with these numbers.

Related Functions:

sAT_PlusCPBW ()

6.4.4 Call-back Functions

6.4.4.1 rAT_PlusCPIN () – Enter PIN

Command Reference:

GSM 07.07 (8.3)

Function Definition:

```
void rAT_PlusCPIN (T_ACI_CPIN_RSLT rslt);
```

Parameters:

name	buffer size	comment	
rslt	---	result code	IN

Return:

none

Function Group:

Phonebook Control

Description:

The rAT_PlusCPIN function is used to return the PIN status of the ME if the request using the qAT_PlusCPIN was not completed immediately. The current PIN status is indicated by the value of *rslt*.

Related Functions:

sAT_PlusCPIN ()

qAT_PlusCPIN ()

6.4.4.2 rAT_PlusCRSM () – Restricted SIM Access

Command Reference:

GSM 07.07 (8.18)

Function Definition:

```
void rAT_PlusCRSM (SHORT sw1, SHORT sw2, SHORT rspLen, UBYTE * rsp);
```

Parameters:

name	buffer size	comment	
sw1	---	SIM result code 1	IN
sw2	---	SIM result code 2	IN
rspLen	---	length of response data in bytes	IN
rsp	---	pointer to response data	IN

Return:

none

Function Group:

SIM access

Description:

The function rAT_PlusCRSM is used to inform the application about the result of a previous request to a SIM field using the function sAT_PlusCRSM. Parameter *sw1* and *sw2* are the SIM result according to GSM 11.11. If response data is available a pointer to the string of bytes *rsp* is provided, the parameter *rspLen* indicates the data length in bytes.

Related Functions:

sAT_PlusCRSM ()

6.5 ME Errors

6.5.1 Call-back Functions

6.5.1.1 rAT_PlusCME () - ME Error Result Code

Command Reference:

GSM 07.07 (9.2)

Function Definition:

```
void rAT_PlusCME (T_ACI_AT_CMD cmdId, T_ACI_CME_ERR err);
```

Parameters:

name	comment	
cmdId	related command identifier	IN
err	error result code	IN

Return:

none

Function Group:

Results

Description:

The rAT_PlusCME function is used to report error result codes upon failures of last executed functions. The *cmdId* value determines the command function to which the error code, represented by *err* is related.

Related Functions:

None

6.6 Functions Diverted from TIA IS-101

6.6.1 Set Functions

6.6.1.1 sAT_PlusVTS () - DTMF Generation

Command Reference:*GSM 07.07 (C.11)***Function Definition:**

```
T_ACI_RETURN sAT_PlusVTS (T_ACI_CMD_SRC srcId, CHAR dtmf, T_ACI_VTS_MOD mode);
```

Parameters:

name	comment	
srcId	command source identifier	IN
dtmf	DTMF tone	IN
mode	DTMF mode	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Audio Functions

Description:

The sAT_PlusVTS function allows the transmission of DTMF tones. These tones may be used (for example) when announcing the start of a recording period. The *dtmf* value represents a single ASCII character. The *mode* value is used to affect the way the DTMF tone is send (automatic, manually start and stop).

Related Functions:

None

7 Functions Diverted from GSM Rec. 07.05

7.1 General Configuration Functions

7.1.1 Set Functions

7.1.1.1 sAT_PlusCSMS () - Select Message Service

Command Reference:

GSM 07.05 (3.2.1)

Function Definition:

T_ACI_RETURN sAT_PlusCSMS (T_ACI_CMD_SRC srcId, T_ACI_CSMS_SERV service);

Parameters:

name	comment	
srcId	command source identifier	IN
service	message service	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCSMS function selects the messaging service given by the value of *service*. The respective call-back function rAT_PlusCSMS returns the types of messages supported by the ME.

Related Functions:

qAT_PlusCSMS ()

rAT_PlusCSMS ()

7.1.1.2 sAT_PlusCPMS () - Preferred Message Storage

Command Reference:

GSM 07.05 (3.2.2)

Function Definition:

T_ACI_RETURN sAT_PlusCPMS (T_ACI_CMD_SRC srcId, T_ACI_SMS_STOR mem1, T_ACI_SMS_STOR mem2, T_ACI_SMS_STOR mem3);

Parameters:

name	comment	
srcId	command source identifier	IN
mem1	memory from which messages are read and deleted	IN
mem2	memory to which writing and sending operations are made	IN
mem3	memory to which received SMs are preferred to be stored	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCPMS selects memory storages. The *mem1* value represents the memory from which messages are read and deleted, the *mem2* value - the memory to which writing and sending operations are made and *mem3* - the memory to which received SMS are stored.

Related Functions:

qAT_PlusCPMS ()
rAT_PlusCPMS ()
rAT_PlusCMS ()
rAT_PlusCME ()
rAT_OK()

7.1.1.3 sAT_PlusCMGF () - SMS Format

Command Reference:

GSM 07.05 (3.2.3)

Function Definition:

T_ACI_RETURN sAT_PlusCMGF (T_ACI_CMD_SRC srcId, T_ACI_CMGF_MOD mode);

Parameters:

name	comment	
srcId	command source identifier	IN
mode	input and output format of messages	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Short Message Services

Description:

The sAT_PlusCMGF function sets the input and output format of messages. The *mode* value indicates the format of messages for the serial port used with send, list, read and write functions and unsolicited calls to call-back functions resulting from received messages. Mode can be either PDU mode (entire TP data units used) or text mode (headers and body of the messages given as separate parameters) .

Related Functions:

qAT_PlusCMGF ()

7.1.2 Query Functions

7.1.2.1 qAT_PlusCSMS () - Select Message Service

Command Reference:

GSM 07.05 (3.2.1)

Function Definition:

```
T_ACI_RETURN qAT_PlusCSMS (T_ACI_CMD_SRC srcId, T_ACI_CSMS_SERV* service,
                             T_ACI_CSMS_SUPP* mt, T_ACI_CSMS_SUPP* mo, T_ACI_CSMS_SUPP* bm);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
service	---	message service	OUT
mt	---	support of mt	OUT
mo	---	support of mo	OUT
bm	---	support of bm	OUT

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	completed successfully
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The qAT_PlusCSMS function returns the current selected message service and the state of support for mobile originated, mobile terminated and broadcast type messages. The returned values are stored in the respective locations provided by *service*, *mo*, *mt* and *bm* if the function returns successfully. If the processing of the request can not be completed immediately, the function returns with the value AT_EXCT, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCSMS.

Related Functions:

```
rAT_PlusCSMS ()
sAT_PlusCSMS ()
rAT_PlusCME ()
rAT_OK()
```

7.1.2.2 qAT_PlusCPMS () - Preferred Message Storage

Command Reference:

GSM 07.05 (3.2.2)

Function Definition:

```
T_ACI_RETURN qAT_PlusCPMS (T_ACI_CMD_SRC srcId);
```

Parameters:

name	buffer size	comment
srcId	---	command source identifier IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The qAT_PlusCPMS function starts querying the selected preferred memory storages (see GSM 07.05 section 3.1 <mem1>, <mem2> and <mem3>), the number of messages currently in each memory and the total number of message locations in each memory. The function returns with the value AT_EXCT, indicating that the request is still being executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCPMS.

Related Functions:

rAT_PlusCPMS ()

sAT_PlusCPMS ()

rAT_PlusCME ()

rAT_OK()

7.1.2.3 qAT_PlusCMGF () - SMS Format

Command Reference:

GSM 07.05 (3.2.3)

Function Definition:

```
T_ACI_RETURN qAT_PlusCMGF (T_ACI_CMD_SRC srcId, T_ACI_CMGF_MOD* mode);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
mode	---	mode of message format	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Short Message Services

Description:

The qAT_PlusCMGF function returns the current setting for the message format used by the serial port. The returned value is stored in the location provided by *mode* if the function returns successfully.

Related Functions:

sAT_PlusCMGF ()

7.1.3 Call-back Functions

7.1.3.1 rAT_PlusCSMS () - Select Message Service

Command Reference:

GSM 07.05 (3.2.1)

Function Definition:

```
void rAT_PlusCSMS (T_ACI_CSMS_SERV service, T_ACI_CSMS_SUPP mt, T_ACI_CSMS_SUPP mo,  
                  T_ACI_CSMS_SUPP bm);
```

Parameters:

name	comment	
service	message service	IN
mt	support of mt	IN
mo	support of mo	IN
bm	support of bm	IN

Return:

none

Function Group:

Short Message Services

Description:

The rAT_PlusCSMS function is used to return the requested setting for the messaging service and the supported message types if the request using the qAT_PlusCSMS was not immediately completed. The requested setting is passed via *service*, *mt*, *mo* and *bc*.

Related Functions:

qAT_PlusCSMS ()

sAT_PlusCSMS ()

7.1.3.2 rAT_PlusCPMS () - Preferred Message Storage

Command Reference:

GSM 07.05 (3.2.2)

Function Definition:

```
void rAT_PlusCPMS (T_ACI_SMS_STOR_OCC* mem1, T_ACI_SMS_STOR_OCC* mem2,  
                  T_ACI_SMS_STOR_OCC* mem3);
```

Parameters:

name	comment	
mem1	setting for mem 1	IN
mem2	setting for mem 2	IN
mem3	setting for mem 3	IN

Return:

none

Function Group:

Short Message Services

Description:

The rAT_PlusCPMS function is used to return the requested settings for the preferred message storages if the request using the qAT_PlusCPMS was not completed immediately. The requested settings are passed via the references for *mem1*, *mem2* and *mem3*.

Related Functions:

qAT_PlusCPMS ()

sAT_PlusCPMS ()

7.1.3.3 rAT_PlusCMS () - Message Service Failure

Command Reference:

GSM 07.05 (3.2.5)

Function Definition:

```
void rAT_PlusCMS (T_ACI_AT_CMD cmdId, T_ACI_CMS_ERR err);
```

Parameters:

name	comment	
cmdId	related command identifier	IN
err	error result code	IN

Return:

none

Function Group:

Results

Description:

The rAT_PlusCMS function is used to report error result codes on failures of the last executed functions regarding short messages. The value of *cmdId* determines the command function to which the error code represented by *err* is related.

Related Functions:

None

7.2 Message Configuration Functions

7.2.1 Set Functions

7.2.1.1 sAT_PlusCSCA () - Service Centre Address

Command Reference:

GSM 07.05 (3.3.1)

Function Definition:

```
T_ACI_RETURN sAT_PlusCSCA (T_ACI_CMD_SRC srcId, CHAR* sca, T_ACI_TOA* tosca);
```

Parameters:

name	comment	
srcId	command source identifier	IN
sca	service center address	IN
tosca	type of service center address	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Short Message Services

Description:

The sAT_PlusCSCA function updates the SMSC address, through which mobile originated SMS are transmitted. The buffer provided by *sca* contains the service center address. The value of *tosca* represents the respective type of address.

Related Functions:

qAT_PlusCSCA ()

7.2.1.2 sAT_PlusCSMP () - Set Text Mode Parameters**Command Reference:**

GSM 07.05 (3.3.2)

Function Definition:

```
T_ACI_RETURN sAT_PlusCSMP (T_ACI_CMD_SRC srcId, SHORT fo, SHORT vprel, T_ACI_VP_ABS* vpabs,
                           SHORT pid, SHORT dcs);
```

Parameters:

name	comment	
srcId	command source identifier	IN
fo	first octet of SMS-DELIVER, SMS-SUBMIT, SMS-STATUS-REPORT or SMS-COMMAND	IN
vprel	TP-Validity-Period in numerical format	IN
vpabs	TP-Validity-Period in string format	IN
pid	TP-Protocol-Identifier	IN
dcs	data coding scheme	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Short Message Services

Description:

The sAT_PlusCSMP function selects values for additional parameters needed when a SM is sent to the network or placed in a storage when text format message mode is selected. It is possible to set the validity period starting from when the SM is received by the SMSC (*vpval*) or to define the absolute time of the validity period termination (*vpstr*). When the *fo* parameter is used, a decision is made whether either *vpval* or *vpstr* is valid.

Related Functions:

qAT_PlusCSMP ()

7.2.1.3 sAT_PlusCSCB () - Select Cell Broadcast Message Types

Command Reference:

GSM 07.05 (3.3.4)

Function Definition:

```
T_ACI_RETURN sAT_PlusCSCB (T_ACI_CMD_SRC srcId, T_ACI_CSCB_MOD mode, USHORT* mids,
                           USHORT* dcscs);
```

Parameters:

name	comment	
srcId	command source identifier	IN
mode	mode of acceptance	IN
mids	possible combinations of CBM message identifiers	IN
dcscs	possible combinations of CBM data coding schemes	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Short Message Services

Description:

The sAT_PlusCSCB function selects which types of CBMs are to be received by the ME. The buffer provided by *mids* contains all of the different possible combinations of CBM message identifiers. The buffer provided by *dcscs* contains all of the different possible combinations of CBM data coding schemes. With use of the parameter *mode*, message types specified by *mids* and *dcscs* are set to either be considered or not.

Related Functions:

qAT_PlusCSCB ()

7.2.1.4 sAT_PlusCSAS () - Save Settings

Command Reference:

GSM 07.05 (3.3.5)

Function Definition:

T_ACI_RETURN sAT_PlusCSAS (T_ACI_CMD_SRC srcId, SHORT profile);

Parameters:

name	comment	
srcId	command source identifier	IN
profile	profile number where settings are to be stored	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCSAS function saves current message service settings to a non-volatile memory and the SIM. All settings specified by function calls to sAT_PlusCSCA, sAT_PlusCSMP and sAT_PlusCSCB are saved.

Related Functions:

sAT_PlusCRES ()
sAT_PlusCSCA ()
sAT_PlusCSMP ()
sAT_PlusCSCB ()
rAT_PlusCMS ()
rAT_PlusCME ()
rAT_OK()

7.2.1.5 sAT_PlusCRES () - Restore Settings

Command Reference:

GSM 07.05 (3.3.6)

Function Definition:

```
T_ACI_RETURN sAT_PlusCRES (T_ACI_CMD_SRC srcId, SHORT profile);
```

Parameters:

name	comment	
srcId	command source identifier	IN
profile	profile number from where settings are to be restored	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCRES function restores message service settings from non-volatile memory and SIM. All settings specified by function calls to sAT_PlusCSCA, sAT_PlusCSMP and sAT_PlusCSCB are restored. The value of *profile* defines the profile to be restored.

Related Functions:

sAT_PlusCSAS ()
sAT_PlusCSCA ()
sAT_PlusCSMP ()
sAT_PlusCSCB ()
rAT_PlusCMS ()
rAT_PlusCME ()
rAT_OK()

7.2.2 Query Functions

7.2.2.1 qAT_PlusCSCA () - Service Centre Address

Command Reference:

GSM 07.05 (3.3.1)

Function Definition:

T_ACI_RETURN qAT_PlusCSCA (T_ACI_CMD_SRC srcId, CHAR* sca, T_ACI_TOA* tosca);

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
sca	MAX_NUM_LEN	service centre address	OUT
tosca	---	type of service centre address	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Short Message Services

Description:

The qAT_PlusCSCA function returns the current setting for the Service Center Address and the corresponding type of address. The returned values are stored in the respective locations provided by *sca* and *tosca* if the function returns successfully.

Related Functions:

sAT_PlusCSCA ()

7.2.2.2 qAT_PlusCSMP () - Set Text Mode Parameters

Command Reference:

GSM 07.05 (3.3.2)

Function Definition:

```
T_ACI_RETURN qAT_PlusCSMP (T_ACI_CMD_SRC srcId, SHORT* fo, SHORT* vprel,  
                           T_ACI_VP_ABS* vpabs, SHORT* pid, SHORT* dcs);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
fo	---	first octet	OUT
vprel	---	relative validity period value	OUT
vpabs	---	absolute validity period string	OUT
pid	---	protocol identifier	OUT
dcs	---	data coding scheme	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Short Message Services

Description:

The qAT_PlusCSMP function returns the current setting for the text mode parameters. The returned values are stored in the respective locations provided by *fo*, *vpval*, *vpstr*, *pid* and *dcs* if the function returns successfully.

Related Functions:

sAT_PlusCSMP ()

7.2.2.3 qAT_PlusCSCB () - Select Cell Broadcast Message Types

Command Reference:

GSM 07.05 (3.3.4)

Function Definition:

```
T_ACI_RETURN qAT_PlusCSCB (T_ACI_CMD_SRC srcId, T_ACI_CSCB_MOD* mode, USHORT* mids,
                           USHORT* dcscs);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
mode	---	mode for cell broadcast messages	OUT
mids	MAX_CBM_TYPES	message identifiers	OUT
dcscs	MAX_CBM_TYPES	data coding schemes	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Short Message Services

Description:

The qAT_PlusCSCB function returns the current setting for the receiving of Cell Broadcast Messages. The returned values are stored in the respective locations provided by *mode*, *mids* and *dcscs* if the function returns successfully.

Related Functions:

sAT_PlusCSCB ()

7.3 Message Receiving and Reading Functions

7.3.1 Set Functions

7.3.1.1 sAT_PlusCNMI () - New Message Indications to TE

Command Reference:

GSM 07.05 (3.4.1)

Function Definition:

```
T_ACI_RETURN sAT_PlusCNMI (T_ACI_CMD_SRC srcId, T_ACI_CNMI_MT mt, T_ACI_CNMI_BM bm,  
                           T_ACI_CNMI_DS ds);
```

Parameters:

name	comment	
srcId	command source identifier	IN
mt	indication routing for SMS-DELIVERs	IN
bm	indication routing for CBMs	IN
ds	indication routing for SMS-STATUS	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Short Message Services

Description:

The sAT_PlusCNMI function selects, how the receiving of new messages from the network is indicated. The *mt* value sets the result code indication routing for SMS-DELIVERs, the *bm* value for CBMs and the *ds* value for SMS-STATUS-REPORTs.

Related Functions:

qAT_PlusCNMI ()

rAT_PlusCMS ()

7.3.1.2 sAT_PlusCMGL () - List Messages

Command Reference:

GSM 07.05 (3.4.2)

Function Definition:

```
T_ACI_RETURN sAT_PlusCMGL (T_ACI_CMD_SRC srcId, T_ACI_SMS_STAT state,  
                           SHORT startIdx, T_ACI_SMS_READ rdMode);
```

Parameters:

name	comment	
srcId	command source identifier	IN
state	message status	IN
startIdx	start index	IN
rdMode	read mode	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCMGL function lists the messages from the preferred message storage location, starting with index *startIdx* and with the status set to *state*. The value of *rdMode* is used to control the setting of the message status after reading.

Related Functions:

rAT_PlusCML ()
rAT_PlusCME ()
rAT_OK()

7.3.1.3 sAT_PlusCMGR () - Read Message

Command Reference:

GSM 07.05 (3.4.3)

Function Definition:

```
T_ACI_RETURN sAT_PlusCMGR (T_ACI_CMD_SRC srcId, UBYTE index, T_ACI_SMS_READ rdMode);
```

Parameters:

name	comment	
srcId	command source identifier	IN
index	message storage location	IN
rdMode	read mode	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCMGR function reads a message from the preferred message storage location *index*. The value of *rdMode* is used to control the setting of the message status after reading.

Related Functions:

rAT_PlusCMR ()

rAT_PlusCME ()

rAT_OK()

7.3.1.4 sAT_PlusCNMA () – New Message Acknowledgement to ME/TA**Command Reference:**

GSM 07.05 (3.4.4)

Function Definition:

```
T_ACI_RETURN sAT_PlusCNMA (T_ACI_CMD_SRC srcId);
```

Parameters:

name	comment	
srcId	command source identifier	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCNMA function acknowledges correct reception of a new message (SMS-DELIVER or SMS-STATUS-REPORT) which is routed directly to the TE. This acknowledgement command (causing ME to send RP-ACK to the network) shall be used when sAT_PlusCSMS parameter *service* equals 1.

Related Functions:

sAT_PlusCNMAPdu ()

sAT_PlusCSMS ()

7.3.2 Query Functions

7.3.2.1 qAT_PlusCNMI () - New Message Indications to TE

Command Reference:

GSM 07.05 (3.4.1)

Function Definition:

```
T_ACI_RETURN qAT_PlusCNMI (T_ACI_CMD_SRC srcId, T_ACI_CNMI_MT* mt, T_ACI_CNMI_BM* bm,  
                           T_ACI_CNMI_DS* ds);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
mt	---	indication mode for mt	OUT
bm	---	indication mode for bm	OUT
ds	---	indication mode for ds	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Short Message Services

Description:

The qAT_PlusCNMI function returns the current settings for the result code indication routing for SMS-DELIVERs, CBMs and SMS-STATUS-REPORTs. The returned values are stored in the locations provided by *mt* for SMS-DELIVERs, *bm* for CBMs and *ds* for SMS-STATUS-REPORTs if the function returns successfully.

Related Functions:

sAT_PlusCNMI ()

7.3.3 Call-back Functions

7.3.3.1 rAT_PlusCBM () - New Message Indication for a Cell Broadcasting Message

Command Reference:

GSM 07.05 (3.4.1)

Function Definition:

```
void rAT_PlusCBM (SHORT sn, SHORT mid, SHORT dcs, UBYTE page, UBYTE pages, T_ACI_CBM_DATA* data);
```

Parameters:

name	comment	
sn	serial number in integer format	IN
mid	message identification in integer format	IN
dcs	data coding scheme in integer format	IN
page	number of this page	IN
pages	number of total pages	IN
data	message data	IN

Return:

none

Function Group:

Results

Description:

The rAT_PlusCBM function is used to report an incoming cell broadcasting message if this option was enabled using the sAT_PlusCNMI function. The incoming message is presented in detailed format using the values *sn*, *mid*, *dcs*, *page* *pages* and *data*.

Related Functions:

sAT_PlusCNMI ()

7.3.3.2 rAT_PlusCBMI () - New Message Indication for a Cell Broadcasting Message

Command Reference:

GSM 07.05 (3.4.1)

Function Definition:

void rAT_PlusCBMI (T_ACI_SMS_STOR mem, UBYTE index);

Parameters:

name	comment	
mem	storage memory	IN
index	index of message	IN

Return:

none

Function Group:

Results

Description:

The rAT_PLusCBMI function is used to report an incoming cell broadcasting message if this option was enabled using the sAT_PlusCNMI function. The incoming message is presented using the values *mem* and *index*, from where the message can be read.

Related Functions:

sAT_PlusCNMI ()

7.3.3.3 rAT_PlusCDS () - New Message Indication for a Status Report**Command Reference:**

GSM 07.05 (3.4.1)

Function Definition:

```
void rAT_PlusCDS (SHORT fo, UBYTE mr, CHAR* ra, T_ACI_TOA* tora, T_ACI_VP_ABS* scts,
                  T_ACI_VP_ABS* dt, UBYTE st);
```

Parameters:

name	comment	
fo	first octet in integer format	IN
mr	TP-message-reference in integer format	IN
ra	TP-recipient-address address-value field in string format	IN
tora	TP-recipient-address type-of-address	IN
scts	TP-service-center-time-stamp in time-string format	IN
dt	TP-discharge-time in time-string format	IN
st	TP-Status in integer format	IN

Return:

none

Function Group:

Results

Description:

The rAT_PLusCDS function is used to report an incoming status report message if this option was enabled using the sAT_PlusCNMI function. The incoming message is presented in detailed format using the values *fo*, *mr*, *dcs*, *ra*, *tora*, *scts*, *dt* and *st*.

Related Functions:

sAT_PlusCNMI ()

7.3.3.4 rAT_PlusCMT () - New Message Indication for Mobile Terminated Messages**Command Reference:**

GSM 07.05 (3.4.1)

Function Definition:

```
void rAT_PlusCMT (CHAR* oa, T_ACI_PB_TEXT* alpha, T_ACI_VP_ABS* scts, T_ACI_TOA* tooa, SHORT fo,
    SHORT pid, SHORT dcs, CHAR* sca, T_ACI_TOA* tosca, T_ACI_SM_DATA* data,
    T_ACI_UDH_DATA* udh);
```

Parameters:

name	comment	
oa	TP-Originating-Address Address-Value field in string format	IN
alpha	alphanumeric representation of originating address	IN
scts	TP-service-Center-time-stamp in time-string format	IN
tooa	TP-originating-address type-of-address	IN
fo	first octet in integer format	IN
pid	TP-protocol-identifier in integer format	IN
dcs	data coding scheme in integer format	IN
sca	service center address address-value field in string format	IN
tosca	service center address type-of-address	IN
data	message data	IN
udh	user data header	IN

Return:

none

Function Group:

Results

Description:

The rAT_PlusCMT function is used to report an incoming mobile terminated message if this option was enabled using the sAT_PlusCNMI function. . The incoming message is presented in detailed format using the values *oa*, *alpha*, *scts*, *tooa*, *fo*, *pid*, *dcs*, *sca*, *tosca*, *data* and *udh*.

Related Functions:

sAT_PlusCNMI ()

7.3.3.5 rAT_PlusCMTI () - New Message Indication for Mobile Terminated Messages

Command Reference:

GSM 07.05 (3.4.1)

Function Definition:

void rAT_PlusCMTI (T_ACI_SMS_STOR mem, UBYTE index);

Parameters:

name	comment	
mem	storage memory	IN
index	index of message	IN

Return:

none

Function Group:

Results

Description:

The rAT_PLusCMTI function is used to report an incoming mobile terminated message if this option was enabled using the sAT_PlusCNMI function. The incoming message is presented using the values *mem* and *index*, from where the message can be read.

Related Functions:

sAT_PlusCNMI ()

7.3.3.6 rAT_PlusCMGL () - List Messages

Command Reference:

GSM 07.05 (3.4.2)

Function Definition:

```
void rAT_PlusCMGL (T_ACI_CMGL_SM* smLst);
```

Parameters:

name	comment	
smLst	list of SMS messages	IN

Return:

none

Function Group:

Short Message Services

Description:

The rAT_PlusCMGL function is used to return the requested list of stored messages with the desired status if the request using the qAT_PlusCMGL was not completed immediately.

Related Functions:

qAT_PlusCMGL ()

7.3.3.7 rAT_PlusCMGR () - Read Messages

Command Reference:

GSM 07.05 (3.4.3)

Function Definition:

```
void rAT_PlusCMGR (T_ACI_CMGR_SM* sm, T_ACI_CMGR_CBM* cbm);
```

Parameters:

name	comment	
sm	short message	IN
cbm	cell broadcasting message	IN

Return:

none

Function Group:

Short Message Services

Description:

The rAT_PlusCMGR function is used to return the requested stored messages with the desired index out of the preferred message storage if the request using the sAT_PlusCMGR was not completed immediately. Depending on the setting for the preferred message storage, the requested message is passed by a reference using *sm* or *cbm*.

Related Functions:

sAT_PlusCMGR ()

7.4 Message Sending and Writing Functions

7.4.1 Set Functions

7.4.1.1 sAT_PlusCMGS () - Send Message

Command Reference:

GSM 07.05 (3.5.1)

Function Definition:

T_ACI_RETURN sAT_PlusCMGS (T_ACI_CMD_SRC srcId, CHAR* da, T_ACI_TOA* toda, CHAR* data,
CHAR* sca, T_ACI_TOA* tosca, SHORT isReply);

Parameters:

name	comment	
srcId	command source identifier	IN
da	destination address	IN
toda	type of destination address	IN
data	message data	IN
sca	service centre address	IN
tosca	type of service centre address	IN
isReply	> 0: set TP-Reply-Path explicitly, EQ 0: clear TP-Reply-Path explicitly	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCMGS function sends a message to the network (SMS-SUBMIT). The message reference value *mr* is returned via the respective call-back function rAT_PlusCMGS on successful message delivery. Values can be used to identify the message upon unsolicited delivery status report result code. The buffer provided by *da* contains the destination address. The value provided by *toda* represents the respective type of address. The buffer provided by *data* contains the message itself. The buffer provided by *sca* contains the service centre address. The buffer provided by *tosca* contains the type of service centre address. The paramter *isReply* sets or clears the TP-Reply-Path.

Related Functions:

rAT_PlusCMGS () rAT_PlusCME ()
rAT_PlusCMS () rAT_OK()
sAT_PlusCMGSPdu ()

7.4.1.2 sAT_PlusCMSS () - Send Message from Storage

Command Reference:

GSM 07.05 (3.5.2)

Function Definition:

```
T_ACI_RETURN sAT_PlusCMSS (T_ACI_CMD_SRC srcId, UBYTE index, CHAR* da, T_ACI_TOA* toda);
```

Parameters:

name	comment	
srcId	command source identifier	IN
index	message storage location	IN
da	destination address	IN
toda	type of destination address	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCMSS function sends a message with location value of *index* from preferred message storage to the network (SMS-SUBMIT). If new recipient address *da* is provided, it is used instead of the one stored with the message. The reference value *mr* is returned via the respective call-back function rAT_PlusCMSS upon successful message delivery. The values can be used to identify message upon unsolicited delivery status report result code. The buffer provided by *da* contains the destination address. The value provided by *toda* represents the respective type of address.

Related Functions:

rAT_PlusCMSS ()

rAT_PlusCMS ()

rAT_PlusCME ()

rAT_OK()

7.4.1.3 sAT_PlusCMGW () - Write Message to Memory**Command Reference:**

GSM 07.05 (3.5.3)

Function Definition:

```
T_ACI_RETURN sAT_PlusCMGW (T_ACI_CMD_SRC srcId,
                           SHORT index,
                           CHAR* address, T_ACI_TOA* toa, T_ACI_SMS_STAT stat, CHAR* data,
                           CHAR* sca, T_ACI_TOA* tosca, SHORT isReply);
```

Parameters:

name	comment	
srcId	command source identifier	IN
index	index of location area to be written	IN
address	destination/originating address	IN
toa	type of destination/originating address	IN
stat	message state	IN
data	message data	IN
sca	service centre address	IN
tosca	type of service centre address	IN
isReply	> 0: set TP-Reply-Path explicitly, EQ 0: clear TP-Reply-Path explicitly	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCMGW function stores a message (either SMS-DELIVER or SMS-SUBMIT) to the preferred memory storage. The parameter *index* gives the memory location of the stored message. The buffer provided by *address* contains the destination address. The value provided by *toa* represents the respective type of address. By default, message status is set to "stored unsent", but the *stat* parameter also allows other status values to be provided. The buffer provided by *data* contains the message itself. The buffer provided by *sca* provides the service centre address. The value provided by *tosca* represents the respective type of address. The parameter *isReply* sets or clears the TP-Reply-Path.

Related Functions:

```
rAT_PlusCMGW ()      rAT_OK()
rAT_PlusCMS ()
rAT_PlusCME ()
```

7.4.1.4 sAT_PlusCMGD () - Delete Message

Command Reference:

GSM 07.05 (3.5.4)

Function Definition:

T_ACI_RETURN sAT_PlusCMGD (T_ACI_CMD_SRC srcId, UBYTE index);

Parameters:

name	comment	
srcId	command source identifier	IN
index	message storage location	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:The sAT_PlusCMGD function deletes message from the preferred message storage location *index*.**Related Functions:**

rAT_PlusCMS ()

rAT_PlusCME ()

rAT_OK()

7.4.1.5 sAT_PlusCMGC () - Send Command**Command Reference:**

GSM 07.05 (3.5.5)

Function Definition:

```
T_ACI_RETURN sAT_PlusCMGC (T_ACI_CMD_SRC srcId, SHORT fo, SHORT ct, SHORT pid, SHORT mn,
                           CHAR* da, T_ACI_TOA* toda, T_ACI_CMD_DATA* data);
```

Parameters:

name	comment	
srcId	command source identifier	IN
fo	first octet of SMS-DELIVER, SMS-SUBMIT, SMS-STATUS-REPORT or SMS-COMMAND	IN
ct	TP-Command-Type	IN
pid	TP-Protocol-Identifier	IN
mn	TP-Message-Number	IN
da	destination address	IN
toda	type of destination address	IN
data	message data	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCMGC function sends a command message to the network (SMS-COMMAND). The format of the message text is fixed to be a sequence of two IRA character-long hexadecimal numbers, which ME convert into 8-bit octets. The message reference value *mr* is returned via the respective call-back function rAT_PlusCMGC upon successful message delivery. The value can be used to identify a message upon unsolicited delivery status report result code.

Related Functions:

```
rAT_PlusCMGC ()
rAT_PlusCMS ()
sAT_PlusCMGCPdu ()
rAT_PlusCME ()
rAT_OK()
```

7.4.2 Call-back Functions

7.4.2.1 rAT_PlusCMGS () - Send Message

Command Reference:

GSM 07.05 (3.5.1)

Function Definition:

```
void rAT_PlusCMGS (UBYTE mr);
```

Parameters:

name	comment	
mr	message reference in integer format	IN

Return:

none

Function Group:

Short Message Services

Description:

The rAT_PlusCMGS function is used to indicate the successful sending of a message invoked by the sAT_PlusCMGS function. A message reference is passed by the *mr* value which is used to identify related status result messages.

Related Functions:

sAT_PlusCMGS ()

7.4.2.2 rAT_PlusCMSS () - Send Message from Storage

Command Reference:

GSM 07.05 (3.5.2)

Function Definition:

```
void rAT_PlusCMSS (UBYTE mr);
```

Parameters:

name	comment	
mr	message reference in integer format	IN

Return:

none

Function Group:

Short Message Services

Description:

The rAT_PlusCMSS function is used to indicate the successful sending of a stored message invoked by the sAT_PlusCMSS function. A message reference is passed by the *mr* value which is used to identify related status result messages.

Related Functions:

sAT_PlusCMSS ()

7.4.2.3 rAT_PlusCMGW () - Write Message to Memory

Command Reference:

GSM 07.05 (3.5.3)

Function Definition:

void rAT_PlusCMGW (UBYTE index);

Parameters:

name	comment	
index	index of written message	IN

Return:

none

Function Group:

Short Message Services

Description:

The rAT_PlusCMGW function is used to indicate the successful storing of an SMS-deliver or SMS-submit message to the preferred memory storage mem2 invoked by the sAT_PlusCMGW function. A message index is passed by the *index* value which is used to identify the stored message.

Related Functions:

sAT_PlusCMGW ()

7.4.2.4 rAT_PlusCMGC () - Send Command

Command Reference:

GSM 07.05 (3.5.5)

Function Definition:

```
void rAT_PlusCMGC (UBYTE mr);
```

Parameters:

name	comment	
mr	message reference in integer format	IN

Return:

none

Function Group:

Short Message Services

Description:

The rAT_PlusCMGC function is used to indicate the successful sending of a command message invoked by the sAT_PlusCMGC function. A message reference is passed by the *mr* value which is used to identify related status result messages.

Related Functions:

sAT_PlusCMGC ()

7.5 PDU Mode

7.5.1 Set Functions

7.5.1.1 sAT_PlusCMGSPdu () - Send Message in PDU Mode

Command Reference:

GSM 07.05 (4.3)

Function Definition:

```
T_ACI_RETURN sAT_PlusCMGSPdu (T_ACI_CMD_SRC srcId,
                                CHAR* sca, T_ACI_TOA* toska, UBYTE msg_type,
                                CHAR* da, T_ACI_TOA* toda, UBYTE pid, UBYTE dcs,
                                UBYTE vp_rel, T_vp_abs* vp_abs, T_ACI_SM_DATA* data);
```

Parameters:

name	comment	
srcId	command source identifier	IN
sca	service centre address	IN
tosca	type of service centre address	IN
msg_type	message type	IN
da	destination address	IN
toda	type of destination address	IN
pid	protocol identifier	IN
dcs	data coding scheme	IN
vp_rel	relative validity period	IN
vp_abs	absolute validity period	IN
data	message data	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCMGSPdu function sends a message to the network (SMS-SUBMIT). The message reference value *mr* is returned via the respective call-back function rAT_PlusCMGS on successful message delivery. This value can be used to identify the message upon unsolicited delivery status report result code. The buffer provided

by *da* contains the destination address. The value provided by *toda* represents the respective type of address. The buffer provided by *data* contains the message itself. The buffer provided by *sca* contains the service centre address. The buffer provided by *tosca* contains the type of service centre address. The value provided by *msg_type* represents the message type. The value provided by *pid* represents the protocol identifier. The value provided by *dcs* represents the data coding scheme. The value provided by *vp_rel* provides the relative validity period, the buffer provided by *vp_abs* contains the absolute validity period. The function is only available if the macro SMS_PDU_SUPPORT is defined in the sources.

Related Functions:

rAT_PlusCMGS ()

rAT_PlusCMS ()

sATplusCMGS ()

rAT_PlusCME ()

rAT_OK()

7.5.1.2 sAT_PlusCMGCPdu () - Send Command in PDU Mode**Command Reference:**

GSM 07.05 (4.5)

Function Definition:

```
T_ACI_RETURN sAT_PlusCMGCPdu (T_ACI_CMD_SRC srcId, CHAR* sca,
                                T_ACI_TOA* tosca, UBYTE msg_type,
                                CHAR* da, T_ACI_TOA* toda, SHORT pid, SHORT msg_num,
                                SHORT sms_cmd, T_ACI_CMD_DATA* data);
```

Parameters:

name	comment	
srcId	command source identifier	IN
sca	service centre address	IN
tosca	type of service centre address	IN
msg_type	message type	IN
da	destination address	IN
toda	type of destination address	IN
pid	protocol identifier	IN
msg_num	message number	IN
sms_cmd	SMS command	IN
data	message data	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCMGCPdu function sends a command message to the network (SMS-COMMAND). The message reference value *mr* is returned via the respective call-back function rAT_PlusCMGC upon successful message delivery. The value can be used to identify a message upon unsolicited delivery status report result code. The function is only available if the macro SMS_PDU_SUPPORT is defined in the sources.

Related Functions:

rAT_PlusCMGC ()	rAT_PlusCME ()
rAT_PlusCMS ()	rAT_OK()
sAT_PlusCMGC ()	

7.5.1.3 sAT_PlusCNMAPdu () – New Message Acknowledgement to ME/TA in PDU Mode

Command Reference:

GSM 07.05 (4.6)

Function Definition:

```
T_ACI_RETURN sAT_PlusCNMAPdu (T_ACI_CMD_SRC srcId, UBYTE n, USHORT tp_fcs);
```

Parameters:

name	comment	
srcId	command source identifier	IN
n	controls type of acknowledgement PDU	IN
tp_fcs	failure cause	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Short Message Services

Description:

The sAT_PlusCNMAPdu function acknowledges the correct reception of a new message (SMS-DELIVER or SMS-STATUS-REPORT) which is routed directly to the TE. The value *n* controls the type of the acknowledgement PDU. The value *tp_fcs* gives the error cause if a RP-ERROR (*n*=2) is sent to the network. The function is only available if the macro SMS_PDU_SUPPORT is defined in the sources.

Related Functions:

sAT_PlusCNMA ()

8 Functions Diverted from ITU-T V.25ter

8.1 Generic TA Control Commands

8.1.1 Set Functions

8.1.1.1 sAT_Z () – Reset to Default Configuration

Command Reference:

ITU-T V.25ter (6.1.1)

Function Definition:

T_ACI_RETURN sAT_Z (T_ACI_CMD_SRC srcId, SHORT value);

Parameters:

name	comment	
srcId	command source identifier	IN
value	configuration number (only 0 is supported)	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Generic TA Control Commands

Description:

The sAT_Z function sets all parameters to their factory defaults. The parameter *value* must be equal to 0, all other values are reserved for manufacturer proprietary use.

Related Functions:

sAT_AndF ()

8.1.1.2 sAT_AndF () – Select Manufacturer Defaults

Command Reference:

ITU-T V.25ter (6.1.2)

Function Definition:

T_ACI_RETURN sAT_AndF (T_ACI_CMD_SRC srcId, SHORT value);

Parameters:

name	comment	
srcId	command source identifier	IN
value	configuration number (only 0 is supported)	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Generic TA Control Commands

Description:

The sAT_AndF function selects the factory-defined configuration. The parameter *value* must be equal to 0, all other values are reserved for manufacturer proprietary use.

Related Functions:

sAT_Z ()

8.1.1.3 sAT_PlusGCI () – Select Country of Installation (not implemented yet)**Command Reference:**

ITU-T V.25ter (6.1.10)

Function Definition:

T_ACI_RETURN sAT_PlusGCI (T_ACI_CMD_SRC srcId, UBYTE country);

Parameters:

name	comment	
srcId	command source identifier	IN
country	country of installation	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Generic TA Control Commands

Description:

The function is used to select the country of installation for the DCE. The value of *country* indicates the country code according to the specification T.35.

Related Functions:

qAT_PlusGCI ()

8.1.2 Query Functions

8.1.2.1 qAT_PlusGCI () – Select Country of Installation (not implemented yet)

Command Reference:

ITU-T V.25ter (6.1.10)

Function Definition:

T_ACI_RETURN qAT_PlusGCI (T_ACI_CMD_SRC srcId, UBYTE*country);

Parameters:

name	comment	
srcId	command source identifier	IN
country	country of installation	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

Generic TA Control Commands

Description:

The qAT_PlusCGI function returns the current setting for the country of installation. The returned value is stored in the respective location provided by *country* if the function returns successfully.

Related Functions:

sAT_PlusGCI ()

8.2 TE-TA Interface Commands

8.2.1 Set Functions

8.2.1.1 sAT_PlusICF () – TE-TA Character Framing

Command Reference:

ITU-T V.25ter (6.2.11)

Function Definition:

T_ACI_RETURN sAT_PlusICF (T_ACI_CMD_SRC srcId, T_ACI_BS_FRM format, T_ACI_BS_PAR parity);

Parameters:

name	comment	
srcId	command source identifier	IN
format	data format	IN
parity	parity setting	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

TE-TA Interface Commands

Description:

The function sAT_PlusICF is used to determine the local serial port start-stop (asynchronous) character framing that the DCE shall use while accepting DTE commands and while transmitting information text and result code.

Related Functions:

qAT_PlusICF ()

8.2.2 Query Functions

8.2.2.1 qAT_PlusICF () – TE-TA Character Framing

Command Reference:

ITU-T V.25ter (6.2.11)

Function Definition:

T_ACI_RETURN qAT_PlusICF (T_ACI_CMD_SRC srcId, T_ACI_BS_FRM* format, T_ACI_BS_PAR* parity);

Parameters:

name	comment	
srcId	command source identifier	IN
format	data format	OUT
parity	parity setting	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

TE-TA Interface Commands

Description:

The function qAT_PlusICF is used to query the ME about the current format and parity settings on the serial interface. If called successfully, the data format is stored in the location *format*, and the parity setting is stored in the location *parity*.

Related Functions:

sAT_PlusICF ()

8.3 Call Control Functions

8.3.1 Set Functions

8.3.1.1 sAT_A () - Answer a Call

Command Reference:

ITU-T V.25ter (6.3.5)

Function Definition:

T_ACI_RETURN sAT_A (T_ACI_CMD_SRC srcId);

Parameters:

name	comment	
srcId	command source identifier	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Call Control

Description:

The sAT_A function is used in two ways. At first the function is used to connect to an incoming call, if no call is currently active. Secondly in case of a combination call, the function is used to switch the call mode from voice to data/fax. If one of the above conditions occur, the function will return with AT_EXCT, indicating that the execution of the command is in progress.

Related Functions:

rAT_NO_CARRIER ()

rAT_CONNECT ()

rAT_OK ()

8.3.1.2 sAT_H () - Hook Control

Command Reference:

ITU-T V.25ter (6.3.6)

Function Definition:

T_ACI_RETURN sAT_H (T_ACI_CMD_SRC srcId);

Parameters:

name	comment	
srcId	command source identifier	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

Call Control

Description:

The sAT_H function instructs the ME

The sAT_H function is used in two ways. At first in case of a combination call the function is used to switch the call mode from data/fax to voice. Secondly the function is used to disconnect from the line, terminating any call in progress. If one of the above conditions occur, the function will return with AT_EXCT, indicating that the execution of the command is in progress.

Related Functions:

rAT_OK ()

8.3.1.3 sAT_0 () – Return to Online Mode**Command Reference:**

ITU-T V.25ter (6.3.7)

Function Definition:

T_ACI_RETURN sAT_0 (T_ACI_CMD_SRC srcId);

Parameters:

name	comment	
srcId	command source identifier	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_EXCT	executing
AT_FAIL	failed

Function Group:

Call Control

Description:

The sAT_0 function instructs the ME to return from command mode to online mode.

Related Functions:

rAT_CONNECT ()

8.3.2 Call-back Functions

8.3.2.1 rAT_BUSY () - Busy Signal Detected

Command Reference:

ITU-T V.25ter (6.3.1)

Function Definition:

```
void rAT_BUSY (T_ACI_AT_CMD cmdId, SHORT cId);
```

Parameters:

name	comment	
cmdID	command identity	IN
cId	call identifier	IN

Return:

none

Function Group:

Results

Description:

The rAT_BUSY function is used to indicate a user busy condition during a mobile originated call setup invoked by the sAT_D function. The value of *cId* identifies the busy call.

Related Functions:

sAT_D ()

8.3.2.2 rAT_CONNECT () - Entering Data Transfer State

Command Reference:

ITU-T V.25ter (6.3.1)

Function Definition:

```
void rAT_CONNECT (T_ACI_AT_CMD cmdId, T_ACI_BS_SPEED speed, SHORT cId);
```

Parameters:

name	comment	
cmdId	related command identifier	IN
speed	data rate	IN
cId	call Id	IN

Return:

none

Function Group:

Results

Description:

The rAT_CONNECT function is used to indicate a connect condition for a data call during a mobile originated or terminated call setup, or an in-call modification. This can be invoked either by the sAT_D or sAT_A functions. The *speed* value indicates the negotiated data rate for the call. The value of *cmdId* determines the function to which the connect indication is related. The value of *cId* identifies the connected call.

Related Functions:

sAT_D ()

sAT_A ()

8.3.2.3 rAT_NO_ANSWER () – Connection Completion Timeout

Command Reference:

ITU-T V.25ter (6.3.1)

Function Definition:

```
void rAT_NO_ANSWER (T_ACI_AT_CMD cmdId, SHORT cId);
```

Parameters:

name	comment	
cmdId	related command identification	IN
cId	call Id	IN

Return:

none

Function Group:

Results

Description:

The rAT_NO_ANSWER function is used to indicate a no answering condition during a mobile originated call setup invoked by the sAT_D function. The value of *cId* identifies the unanswered call.

Related Functions:

sAT_D ()

8.3.2.4 rAT_NO_CARRIER () – Connection Terminated

Command Reference:

ITU-T V.25ter (6.3.1)

Function Definition:

```
void rAT_NO_CARRIER (T_ACI_AT_CMD cmdId, SHORT cId);
```

Parameters:

name	comment	
cmdId	related command identification	IN
cId	call Id	IN

Return:

none

Function Group:

Results

Description:

The rAT_NO_CARRIER function is used to indicate an unsuccessful call establishment during a mobile originated or terminated call setup invoked by the sAT_D or sAT_A function or a mobile terminated release of the call. The value of *cId* identifies the terminated call.

Related Functions:

sAT_D ()

sAT_A ()

8.3.2.5 rAT_OK () - Command Successfully Completed

Command Reference:

ITU-T V.25ter (6.3.1)

Function Definition:

```
void rAT_OK (T_ACI_AT_CMD cmdId);
```

Parameters:

name	comment	
cmdId	related command identification	IN

Return:

none

Function Group:

Results

Description:

The rAT_OK function is used to report a successful completion of the last executed functions. The *cmdId* value determines the function to which the successful status is related.

Related Functions:

None

8.4 Data Compression Functions

8.4.1 Set Functions

8.4.1.1 sAT_PlusDS () – Data Compression

Command Reference:

ITU-T V.25ter (6.6.1)

Function Definition:

```
T_ACI_RETURN sAT_Plus_DS (T_ACI_CMD_SRC srcId, T_ACI_DS_DIR dir, T_ACI_DS_COMP comp,
                           LONG maxDict, SHORT maxStr);
```

Parameters:

name	comment	
srcId	command source identifier	IN
dir	desired direction of data compression	IN
comp	mode of continue	IN
maxDict	maximum number of dictionary entries	IN
maxStr	maximum string length	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Data Compression

Description:

The functions sAT_PlusDS is used to set parameters for the V.42bis data compression. The value *dir* determines the desired direction for data compression. The value *comp* specifies whether or not the DCE should continue to operate if the desired result is not obtained. The value *maxDict* specifies the maximum number of dictionary entries, which should be negotiated. The value *maxStr* specifies the maximum string length to be negotiated.

Related Functions:

qAT_PlusDS ()

8.4.2 Query Functions

8.4.2.1 qAT_PlusDS () – Data Compression

Command Reference:

ITU-T V.25ter (6.6.1)

Function Definition:

T_ACI_RETURN qAT_Plus_DS (T_ACI_CMD_SRC srcId, T_ACI_DS_DIR* dir, T_ACI_DS_COMP* comp,
LONG *maxDict, SHORT *maxStr);

Parameters:

name	comment	
srcId	command source identifier	IN
dir	desired direction of data compression	OUT
comp	compression negotiation	OUT
maxDict	maximum number of dictionary entries	OUT
maxStr	maximum string length	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

Data Compression

Description:

The qAT_PlusDS function queries the settings for V.42bis data compression. If called successfully, the desired direction of data compression is stored in the location *dir*. The compression negotiation, which determines whether or not the DCE should continue to operate if the desired result is not obtained, is stored in the location *comp*. The maximum number of dictionary entries is stored in the location *maxDict*, and the maximum string length to be negotiated is stored in the location *maxStr*.

Related Functions:

sAT_PlusDS ()

8.4.3 Call-back Functions:

8.4.3.1 rAT_PlusDR () Data Compression

Command Reference:

ITU-T V.25ter(6.6.2)

Function Definition:

```
void rAT_PlusDR (T_ACI_DR_TYP type);
```

Parameters:

name	comment	
type	direction of data compression	IN

Return:

none

Function Group:

Data Compression

Description:

The rAT_PlusDR function reports the current data compression settings.

Related Functions:

sAT_PlusDS ()

9 Miscellaneous

9.1.1 Set Functions

9.1.1.1 sAT_Abort () Abort execution for pending command

Command Reference:

none

Function Definition:

```
T_ACI_RETURN sAT_Abort (T_ACI_CMD_SRC srcId, T_ACI_AT_CMD cmd);
```

Parameters:

name	comment	
srcId	command source identifier	IN
cmd	related command	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_EXCT	executing
AT_FAIL	failed

Function Group:

none

Description:

sAT_Abort aborts any abortable pending command identified by the value of *cmd*.

Related functions:

9.1.1.2 sAT_PercentNRG () – Set Registration Mode and Service Mode**Command Reference:**

none

Function Definition:

```
T_ACI_RETURN sAT_PercentNRG (T_ACI_CMD_SRC srcId, T_ACI_NRG_RGMD regMode,
                              T_ACI_NRG_SVMD srvMode, T_ACI_NRG_FRMT oprFmt, CHAR* opr);
```

Parameters:

name	comment	
srcId	command source identifier	IN
regMode	registration mode	IN
srvMode	service mode	IN
oprFmt	format for network operator parameter	IN
opr	network operator	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	completed successfully
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

none

Description:

The sAT_PercentNRG function sets the registration and service mode. The value *regMode* selects manual or automatic registration. The value *oprFmt* selects the format for the network operator parameter *opr*, which is used if manual registration is requested. The value *opr* itself contains the location of a buffer, which describes the operator either in numerical or alphanumeric format. The value *srvMode* selects the desired service mode to be registered to.

Related functions:

qAT_PercentNRG ()
sAT_PLusCOPS ()

9.1.1.3 sAT_PercentPBCF () – Set Phone Configuration

Command Reference:

none

Function Definition:

```
T_ACI_RETURN sAT_PercentPBCF (T_ACI_CMD_SRC srcId, T_ACI_PBCF_LDN ldn, T_ACI_PBCF_LRN lrn);
```

Parameters:

name	comment	
srcId	command source identifier	IN
ldn	last dialed number configuration	IN
lrn	last received number configuration	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

none

Description:

The sAT_PercentPBCF function sets the last dialed and/or last received number configuration. The automatic storing of the last dialed and/or last received number can be enabled and disabled with this function.

Related functions:

qAT_PercentPBCF ()

9.1.1.4 sAT_PercentPVRF () – Verify a Specific Pin**Command Reference:**

none

Function Definition:

```
T_ACI_RETURN sAT_PercentVRF (T_ACI_CMD_SRC srcId, T_ACI_PVRF_TYPE type,
                              CHAR* pin, CHAR* newpin);
```

Parameters:

name	comment	
srcId	command source identifier	IN
type	type of pin (PINn/PUKn)	IN
pin	string of PIN chars	IN
newpin	string of PIN chars required if requested PIN is SIM PUK	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

none

Description:

The sAT_PercentPVRF function verifies a PIN or PUK. The value of *type* represents the type of the PIN or PUK. The value *pin* provides the buffer location of the PIN or PUK. If *type* is PVRV_TYPE_Puk1 or PVRF_TYPE_Puk2, the parameter *newpin* is required and a call of this function will cause the setting of a new PIN1 or PIN2 according to the value *type* and the content of the buffer *newpin*. The function returns with the value AT_EXCT, indicating that the request is still being processed.

Related functions:

```
qAT_PercentPVRF ()
sAT_PlusCPIN ()
rAT_PlusCME ()
rAT_OK ()
```

9.1.1.5 sAT_PercentSATC () – Configure SIM Toolkit Profile**Command Reference:**

none

Function Definition:

T_ACI_RETURN sAT_PercentSATC (T_ACI_CMD_SRC srcId, SHORT len, UBYTE* satCnfg);

Parameters:

name	comment	
srcId	command source identifier	IN
len	length of satCnfg	IN
satCnfg	SIM application toolkit configuration	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

none

Description:

The sAT_PercentSATC function sends STK profile to SIM to indicate supported features by ME. This has to be done prior to the first SIM activation request. The value *len* describes the length of the profile, *satCnfg* is the location where the profile is located.

Related functions:

sAT_PercentSATR ()
sAT_PercentSATE ()
sAT_PercentSATT ()
rAT_PercentSATI ()
rAT_PercentSATE ()
rAT_PercentSATN ()
rAT_PercentSATA ()

9.1.1.6 sAT_PercentSATE () – Envelope SIM Toolkit Command**Command Reference:**

none

Function Definition:

T_ACI_RETURN sAT_PercentSATE (T_ACI_CMD_SRC srcId, SHORT len, UBYTE* satCmd);

Parameters:

name	comment	
srcId	command source identifier	IN
len	length of satCmd	IN
satCmd	SIM application toolkit command	IN

Return:

symbolic constant	comment
AT_EXCT	executing
AT_FAIL	failed
AT_BUSY	command handler is busy

Function Group:

none

Description:

The sAT_PercentSATE function sends an STK command to the SIM using the envelope format. The location *satCmd* contains the STK command, the value *len* describes the respective length of this command.

Related functions:

sAT_PercentSATC ()
 sAT_PercentSATR ()
 sAT_PercentSATT ()
 rAT_PercentSATI ()
 rAT_PercentSATE ()
 rAT_PercentSATN ()
 rAT_PercentSATA ()

9.1.1.7 sAT_PercentSATR () – Response on STK Command**Command Reference:**

none

Function Definition:

```
T_ACI_RETURN sAT_PercentSATR (T_ACI_CMD_SRC srcId, SHORT len, UBYTE* satCmd);
```

Parameters:

name	comment	
srcId	command source identifier	IN
len	length of satCmd	IN
satCmd	SIM application toolkit command	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

none

Description:

The sAT_PercentSATR function is used to response to a former STK command sent by SIM application. The location *satCmd* contains the STK response, the value *len* describes the respective length of this response.

Related functions:

sAT_PercentSATC ()
sAT_PercentSATE ()
sAT_PercentSATT ()
rAT_PercentSATI ()
rAT_PercentSATE ()
rAT_PercentSATN ()
rAT_PercentSATA ()

9.1.1.8 sAT_PercentSATT () – Sim Application Toolkit Termination**Command Reference:**

none

Function Definition:

```
T_ACI_RETURN sAT_PercentSATT (T_ACI_CMD_SRC srcId, T_ACI_SATT_CS cause);
```

Parameters:

name	comment	
srcId	command source identifier	IN
cause	cause for termination	IN

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

none

Description:

The sAT_PercentSATT function sends STK response with given cause to SIM. This function is used to terminate a SAT session or request. The value *cause* specifies the reason of the termination, e.g. the user clears redialling or redialling exceeds.

Related functions:

sAT_PercentSATC ()
sAT_PercentSATR ()
sAT_PercentSATE ()
rAT_PercentSATI ()
rAT_PercentSATE ()
rAT_PercentSATN ()
rAT_PercentSATA ()

9.1.2 Query Functions

9.1.2.1 qAT_ErrDesc () - Query Error Description of last Error

Command Reference:

none

Function Definiton:

```
T_ACI_ERR_DESC qAT_ErrDesc (void);
```

Parameters:

none

Return:

none

Function Group:

none

Description:

In case the call to a function results in the return code AT_FAIL, a call to this function will return a more detailed description of the error cause. The error description value consists of an error class and an error number. Both values can be disassembled out of the compound error description value using the macros ACI_ERR_DESC_CLASS and ACI_ERR_DESC_NR.

Related functions:

9.1.2.1.1 qAT_PercentCACM () - Query Accumulated Call Meter using PUCT**Command Reference:**

none

Function Definition:

```
T_ACI_ERR_DESC qAT_PercentCACM (T_ACI_CMD_SRC srcId, CHAR* cur, CHAR* val);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
cur	MAX_CUR_LEN	currency	OUT
val	MAX_NUM_LEN	value	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

none

Description:

The function qAT_PercentCACM queries the accumulated call meter using the price per unit currency table. If the function returns successfully, the returned values are stored in the locations given by *cur* and *val*.

Related functions:

qAT_PercentCAOC ()

qAT_PlusCACM ()

9.1.2.2 qAT_PercentCAL () - Query Current Call Status

Command Reference:

none

Function Definiton:

```
T_ACI_ERR_DESC qAT_PercentCAL (T_ACI_CMD_SRC srcId, T_ACI_CAL_ENTRY* callTable);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
callTable	MAX_CALL_NR	call table	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

none

Description:

The function qAT_PercentCAL queries the current call status of all current calls. The returned entries are stored in the location provided by the value *callTable* if the function returns successfully.

Related functions:

qAT_PlusCLCC ()

9.1.2.3 qAT_PercentCAOC () - Query Current Call Meter using PUCT

Command Reference:

none

Function Definiton:

```
T_ACI_ERR_DESC qAT_PercentCAOC (T_ACI_CMD_SRC srcId, CHAR* curr, CHAR* val);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
cur	MAX_CUR_LEN	currency	OUT
val	MAX_NUM_LEN	value	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

none

Description:

The function qAT_PercentCAOC queries the current call meter using the price per unit currency table. If the function returns successfully, the returned values are stored in the locations given by *cur* and *val*.

Related functions:

qAT_PercentCACM ()

qAT_PlusCAOC ()

9.1.2.4 qAT_PercentCOLR () - Query Setting for Connected Line Restriction Supplementary Service**Command Reference:**

none

Function Definiton:

```
T_ACI_ERR_DESC qAT_PercentCOLR (T_ACI_CMD_SRC srcId, T_ACI_COLR* stat);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
stat	---	connected line restriction state	OUT

Return:

symbolic constant	comment
AT_EXCT	executing
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

none

Description:

The function qAT_PercentCOLR starts quering the current connected line restriction supplementary service status. If called successfully, the value is stored in the location provided by the value *stat*. If the processing of the request can not be processed immediately, the function returns with the value AT_EXT, indicating that the request is still beeing executed. The final result will be passed to the caller via the respective call-back function rAT_PlusCOLR.

Related functions:

rAT_PlusCOLR ()

9.1.2.5 qAT_PercentCTV () - Query Current Call Timer

Command Reference:

none

Function Definiton:

```
T_ACI_ERR_DESC qAT_PercentCOLR (T_ACI_CMD_SRC srcId, LONG* ctv);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
ctv	---	call timer value	OUT

Return:

symbolic constant	comment
AT_CMPL	successfully completed
AT_FAIL	failed

Function Group:

none

Description:

The function qAT_PercentCTV queries the current call timer. If the function returns successfully, the current call timer value is stored in the location provided by *ctv*.

Related functions:

9.1.2.6 qAT_PercentNRG () - Query Current Registration Settings**Command Reference:**

none

Function Definiton:

```
T_ACI_ERR_DESC qAT_PercentNRG (T_ACI_CMD_SRC srcId, T_ACI_NRG_RGMD* regMode,
                                T_ACI_NRG_SVMD* srvMode, T_ACI_NRG_FRMT* oprFrmt,
                                T_ACI_NRG_SVMD* srvStat, CHAR* oper);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
regMode	---	registration mode	OUT
srvMode	---	service mode	OUT
oprFrmt	---	operator format	OUT
srvStat	---	service status	OUT
oper	MAX_ALPHA_OPER_LEN	operator	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

none

Description:

The function qAT_PercentNRG queries the current registration settings. On successfull call, the registration mode is stored in the location provided by the value *regMode*. The desired service mode is stored in the location provided by the value *srvMode*. The operator format (numeric or alphanumeric) is stored in the location provided by the value *oprFrmt*. The actual service status is stored in the location provided by the value *srvStat*. The current network operator is stored in the buffer location provided by the value *oper*.

Related functions:

sAT_PercentNRG ()

9.1.2.7 qAT_PercentPBCF () – Query Phone Configuration

Command Reference:

none

Function Definition:

```
T_ACI_RETURN sAT_PercentPBCF (T_ACI_CMD_SRC srcId, T_ACI_PBCF_LDN* ldn, T_ACI_PBCF_LRN* lrn);
```

Parameters:

name	comment	
srcId	command source identifier	IN
ldn	last dialed number configuration	IN
lrn	last received number configuration	IN

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

none

Description:

The qAT_PercentPBCF function queries the last dialed and last received number configuration. If the function returns successfully, the last dialed number configuration is stored in the location provided by *ldn* and the last received number configuration is stored in the location provided by *lrn*.

Related functions:

sAT_PercentPBCF ()

9.1.2.8 qAT_PercentPVRF () - Query the Current Counter for PIN and PUK**Command Reference:**

none

Function Definiton:

```
T_ACI_ERR_DESC qAT_PercentPVRF (T_ACI_CMD_SRC srcId,
                                SHORT* pn1Cnt, SHORT* pn2Cnt,
                                SHORT* pk1Cnt, SHORT*pk2Cnt,
                                T_ACI_PVRF_STAT* ps1,
                                T_ACI_PVRF_STAT* ps2);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
pn1Cnt	---	PIN1 count	OUT
pn2Cnt	---	PIN2 count	OUT
pk1Cnt	---	PUK1 count	OUT
pk2Cnt	---	PUK2 count	OUT
ps1	---	PIN1 status	OUT
ps2	---	PIN2 status	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

none

Description:

The function qAT_PercentPVRF is used to query the reattempt counts of PIN1, PIN2, PUK1 and PUK2 and the status of PIN1 and PIN2. If called successfully, the PIN1 reattempt counter is stored in the location *pn1Cnt* and the PIN2 reattempt counter is stored in the location *pk1Cnt*. The PIN1 status is stored in the location *ps2*. Analogous the same is true for PIN2 and PUK2.

Related functions:

sAT_PercentPVRF ()

9.1.2.9 qAT_PercentRPCT () - Query Raw SIM Data for PUCT**Command Reference:**

none

Function Definiton:

```
T_ACI_ERR_DESC qAT_PercentRPCT (T_ACI_CMD_SRC srcId, T_ACI_RPCT_VAL* rpuct);
```

Parameters:

name	buffer size	comment	
srcId	---	command source identifier	IN
rpuct	---	raw puct data	OUT

Return:

symbolic constant	comment
AT_CMPL	completed successfully
AT_FAIL	failed

Function Group:

none

Description:

The function qAT_PercentRPCT queries the SIM for raw PUCT (Price per Unit and Currency Table) data. If called successfully, the location *rpuct* is filled with the structure elements currency, eppu (elementary price per unit), exp and sexp (sign of exp). For further details, see recommendation GSM 11.11 chapter 10.3.12.

Related functions:

9.1.3 Call-back Functions

9.1.3.1 rAT_PercentCLOG () – Command Logging

Command Reference:

none

Function Definition:

```
void rAT_PercentCLOG(T_ACI_CLOG* cmd);
```

Parameters:

name	comment	
cmd	pointer to info structure about logged command	IN

Return:

none

Function Group:

none

Description:

The function rAT_PercentCLOG is called to inform the application about significant commands given at the V.24 which change mobile state in a way which should be displayed to the user by MMI.

Related functions:

```
rAT_PercentRLOG ()
```

9.1.3.2 rAT_PercentCOLR () - Connected Line Identification Restriction

Command Reference:

none

Function Definition:

```
void rAT_PercentCOLR ( T_ACI_COLR_STAT stat);
```

Parameters:

name	comment	
stat	COLR network service status	IN

Return:

none

Function Group:

Supplementary Services

Description:

The rAT_PercentCOLR function is used to return the requested COLR parameter if the request using the qAT_PercentCOLR was not completed immediately. The requested COLR parameter is passed via *stat*.

Related Functions:

qAT_PercentCOLR ()

9.1.3.3 rAT_PercentCPI () – Call Progress Information

Command Reference:

none

Function Definition:

```
void rAT_PercentCPI (SHORT cId, T_ACI_CPI_MSG msgType, T_ACI_CPI_IBT ibt, T_ACI_CPI_TCH tch);
```

Parameters:

name	comment	
cId	call identity	IN
msgType	source of information	IN
ibt	in-band tones indication	IN
tch	traffic channel indication	IN

Return:

none

Function Group:

none

Description:

The function `rAT_PercentCPI` is called to inform the application about the call progress. The value `cId` is the call identification to distinguish between parallel calls. The value `msgType` contains the received layer 3 call control message type. The value `ibt` informs about the availability of in-band tone signalling. The value `tch` informs the application whether a traffic channel has been assigned.

Related functions:

9.1.3.4 rAT_PercentCTV () – Call Timer**Command Reference:**

none

Function Definition:

void rAT_PercentCTV (void);

Parameters:

none

Return:

none

Function Group:

none

Description:

The function rAT_PercentCTV is called to inform the application about a call timer timeout, which occurs every second. The current and accumulated call timer was increased.

Related functions:

9.1.3.5 rAT_PercentKSIR () – Keystroke Interaction Result

Command Reference:

none

Function Definition:

```
void rAT_PercentKSIR (T_ACI_KSIR* ksStat);
```

Parameters:

name	comment	
ksStat	struct to hold various responses caused by keystroke sequences	IN

Return:

none

Function Group:

none

Description:

The function rAT_PercentKSIR is called if the network (or ME, in the special case of an IMEI query) responses on a previous supplementary service transaction. Information regarding the network (or ME) response is found in the union *ir*, the value ksdCmd defines the respective struct of this union. See type declaration of T_ACI_KSIR for further details.

Related functions:

sAT_Dn ()

sAT_Dm ()

9.1.3.6 rAT_PercentRLOG () – Result Logging

Command Reference:

none

Function Definition:

```
void rAT_PercentRLOG (T_ACI_RLOG* rslt);
```

Parameters:

name	comment	
rslt	result paramters of previous command	IN

Return:

none

Function Group:

none

Description:

The function rAT_PercentRLOG is called to inform the application about the result of a previous command given at the V.24, which should be displayed, to the user by the MMI.

Related functions:

rAT_PercentCLOG ()

9.1.3.7 rAT_PercentSATA () – Alert about Sim Application Toolkit Pending Call Setup**Command Reference:**

none

Function Definition:

```
void rAT_PercentSATA (SHORT cId, LONG rdTimeout_ms);
```

Parameters:

name	comment	
cId	call id	IN
rdTimeout_ms	redial timeout in ms	IN

Return:

none

Function Group:

none

Description:

The function rAT_PercentSATA informs the application about a call setup, which is initiated by the SIM application toolkit. The value *cId* represents the call identifier. The value *rdTimeout_ms* is the redial timeout in units of ms. The call is accepted by using the AT A command (sAT_A function) and is rejected by using the AT H command (sAT_H function).

Related functions:

sAT_A ()
sAT_H ()
sAT_PercentSATC ()
sAT_PercentSATR ()
sAT_PercentSATE ()
sAT_PercentSATT ()
rAT_PercentSATI ()
rAT_PercentSATE ()
rAT_PercentSATN ()

9.1.3.8 rAT_PercentSATE () – Notify Application about Result of Envelope Command

Command Reference:

none

Function Definition:

```
void rAT_PercentSATE (SHORT len, UBYTE* satCmd);
```

Parameters:

name	buffer size	comment	
len	---	length	IN
satCmd	MAX_SAT_CMD_LEN	SIM application toolkit command	IN

Return:

none

Function Group:

none

Description:

The function rAT_PercentSATE notifies the application about a result to a previous envelope command sent by the function sAT_PercentSATE. The buffer *satCmd* contains the result itself, the value *len* indicates the respective length.

Related functions:

- sAT_PercentSATC ()
- sAT_PercentSATR ()
- sAT_PercentSATE ()
- sAT_PercentSATT ()
- rAT_PercentSATI ()
- rAT_PercentSATN ()
- rAT_PercentSATA ()

9.1.3.9 rAT_PercentSATI () – Indicate SAT Command Sent by SIM

Command Reference:

none

Function Definition:

```
void rAT_PercentSATI (SHORT len, UBYTE* satCmd);
```

Parameters:

name	buffer size	comment	
len	---	length	IN
satCmd	MAX_SAT_CMD_LEN	SIM application toolkit command	IN

Return:

none

Function Group:

none

Description:

The function `rAT_PercentSATI` is called by the ACI to indicate a SAT command sent by the SIM card to the application. The buffer `satCmd` contains the SIM application toolkit command, the value `len` indicates the respective length.

Related functions:

- sAT_PercentSATC ()
- sAT_PercentSATR ()
- sAT_PercentSATE ()
- sAT_PercentSATT ()
- rAT_PercentSATE ()
- rAT_PercentSATN ()
- rAT_PercentSATA ()

9.1.3.10 rAT_PercentSATN () – Notify Application while Call Setup or Supplementary Service**Command Reference:**

none

Function Definition:

void rAT_PercentSATN (SHORT len, UBYTE* satCmd);

Parameters:

name	buffer size	comment	
len	---	length	IN
satCmd	MAX_SAT_CMD_LEN	SIM application toolkit command	IN

Return:

none

Function Group:

none

Description:

The function rAT_PecentSATN indicates a notification to the MMI. This function is called when a call setup or send SS operation is in progress, when call control by SIM is invoked and when ACI sends terminal responses to the SIM. The buffer *satCmd* contains the SIM application toolkit command, the value *len* indicates the respective length.

Related functions:

sAT_PercentSATC ()
sAT_PercentSATR ()
sAT_PercentSATE ()
sAT_PercentSATT ()
rAT_PercentSATI ()
rAT_PercentSATE ()
rAT_PercentSATA ()

9.1.3.11 rAT_PercentSIMINS () – SIM Inserted**Command Reference:**

none

Function Definition:

void rAT_PercentSIMINS (void);

Parameters:

none

Return:

none

Function Group:

none

Description:

The function rAT_PercentSIMINS is called to indicate that the SIM was inserted.

Related functions:

rAT_PercentSIMREM ()

9.1.3.12 rAT_PercentSIMREM () – SIM Removed**Command Reference:**

none

Function Definition:

void rAT_PercentSIMREM (void);

Parameters:

none

Return:

none

Function Group:

none

Description:

The function rAT_PercentSIMREM is called to indicate that the SIM was removed.

Related functions:

rAT_PercentSIMINS ()

10 Example

The following describes an example for using the functional interface of the ACI entity. This example is a scenario of how a registration after a power-on of the mobile station can be activated from an application. In this special case, a PIN input is required prior to the network registration.

This sequence chart shows the function calls for interaction between an application (MMI) and the ACI entity using the functional interface. Comments are made to clarify the meaning of the commands.

