

Partners can get access to this document by registering at the Mitel Approval Partner Program (A2P2)

Please contact our A2P2 team by email - a2p2.support@mitel.com - for registration or using your account.

SIP-DECT OM Application XML Interface valid

OCTOBER 2015

REQ-0384 V 6.1

SPECIFICATION



NOTICE

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Networks™ Corporation (MITEL®). The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes.

No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

TRADEMARKS

Mitel is a trademark of Mitel Networks Corporation.

Linux® is a registered trademark of Linus Torvalds.

Red Hat® is a registered trademark of Red Hat, Inc.

Java™ is a registered trademark of Oracle Corporation.

Windows® is a registered trademark of Microsoft Corporation.

Other product names mentioned in this document may be trademarks of their respective companies and are hereby acknowledged.

SIP-DECT OM Application XML Interface

Release 6.1

October 2015

Document Version aad-0384 V 6.1

®,™ Trademark of Mitel Networks Corporation

© Copyright 2015 Mitel Networks Corporation

All rights reserved

TABLE OF CONTENTS

1	INTRODUCTION.....	7
1.1	SUMMARY	7
2	PROTOCOL MECHANISMS	8
2.1	CONNECTION STATES	8
2.2	SESSION MANAGEMENT	9
2.3	MESSAGE TRANSPORT AND ENCODING	9
2.4	ENCRYPTION	9
2.5	PERMISSION.....	10
2.6	GENERAL RULES	10
3	COMMON DATA TYPES	11
3.1	REQUESTS.....	11
3.2	RESPONSES.....	11
3.3	ERROR CODES	11
3.4	PROTOCOLTYPE.....	12
3.4.1	PEMCERTIFICATETYPE.....	13
3.4.2	PEMCERTIFICATELISTTYPE	13
3.5	URLTYPE	13
3.6	PERMISSIONTYPE.....	16
3.7	DATETIMETYPE	17
3.8	HEALTHTYPE.....	17
3.9	PPUSERCALLSTATETYPE.....	21
3.10	PPUSERSIPEVENTTYPE	22
3.11	ACCOUNTSTATETYPE.....	22
3.12	VERSIONTYPE.....	22
3.13	PUBLICKEYTYPE	30
4	MESSAGES	31
4.1	BASIC REQUESTS	31
4.1.1	OPEN	31
4.1.2	GETVERSIONS	36
4.1.3	LIMITS	58
4.1.4	SUBSCRIBE.....	60
4.1.5	PING	66

4.1.6	GETSTBSTATE.....	66
4.1.7	EVENTSTBSTATECHANGE	67
4.1.8	EVENTPERMISSIONCHANGE	67
4.1.9	GETPUBLICKEY	67
4.2	LICENSE MANAGEMENT.....	68
4.2.1	LICENSETYPE	68
4.2.2	LICENSERFPTYPE	68
4.2.3	GETLICENSE.....	68
4.2.4	EVENTLICENSECNF.....	70
4.2.5	GETLICENSEDCODECLINES	71
4.2.6	EVENTLICENSEDCODECLINES	71
4.3	LICENSE SERVER CONFIGURATION AND MAINTENANCE	71
4.3.1	LICENSESERVERTYPE.....	71
4.3.2	LICENSESERVERLISTTYPE	72
4.3.3	GETLICENSESERVERLIST.....	72
4.3.4	SETLICENSESERVERLIST	72
4.3.5	EVENTLICENSESERVERLISTCNF.....	73
4.3.6	EVENTLICENSEFILE.....	73
4.4	ACCOUNT CONFIGURATION AND MAINTENANCE.....	73
4.4.1	ACCOUNTTYPE	74
4.4.2	GETACCOUNTSUMMARY	74
4.4.3	GETACCOUNT.....	75
4.4.4	CREATEACCOUNT	75
4.4.5	DELETEACCOUNT	76
4.4.6	SETACCOUNT	76
4.4.7	EVENTACCOUNTCNF.....	77
4.4.8	EVENTACCOUNTSUMMARY.....	77
4.5	OMM SOFTWARE CONFIGURATION AND MANAGEMENT	77
4.5.1	SOFTWAREUPDATE.....	77
4.5.2	GETSOFTWAREUPDATE	78
4.5.3	EVENTSOFTWAREUPDATECNF	79
4.5.4	SYSTEMRESTART.....	79
4.5.5	OMM SOFTWARE IMAGE MANAGEMENT	80

4.5.5.1	GetSoftwareImageURL.....	80
4.5.5.2	SetSoftwareImageURL.....	80
4.5.5.3	EventSoftwareImageURLCnf.....	80
4.5.6	DECT PHONE FIRMWARE MANAGEMENT	80
4.5.6.1	DECT phoneFirmwareUpdateStatusType	80
4.5.6.2	GetPPFirmwareUpdate.....	81
4.5.6.3	SetPPFirmwareUpdate	81
4.5.6.4	GetPPFirmwareUpdateOverview.....	82
4.5.6.5	GetPPFirmwareUpdateStatus	83
4.5.6.6	EventPPFirmwareUpdateCnf.....	83
4.5.6.7	EventPPFirmwareUpdateOverview	84
4.5.6.8	EventPPFirmwareUpdateStatus	84
4.5.6.9	GetPPFirmwareURL.....	84
4.5.6.10	SetPPFirmwareURL	85
4.5.6.11	EventPPFirmwareURLCnf	85
4.6	SYSTEM NAME CONFIGURATION AND MAINTENANCE	85
4.6.1	GETSYSTEMNAME	85
4.6.2	SETSYSTEMNAME.....	86
4.6.3	EVENTSYSTEMNAMECNF	86
4.7	EULA MANAGEMENT	86
4.7.1	GETEULACONFIRM	86
4.7.2	SETEULACONFIRM.....	86
4.7.3	EVENTEULACONFIRMCNF	87
4.8	REMOTE ACCESS CONFIGURATION AND MAINTENANCE	87
4.8.1	GETREMOTEACCESS	87
4.8.2	SETREMOTEACCESS.....	87
4.8.3	EVENTREMOTEACCESSCNF	88
4.9	SYSLOG SERVER CONFIGURATION AND MAINTENANCE.....	88
4.9.1	GETSYSLOGSERVER.....	88
4.9.2	SETSYSLOGSERVER	89
4.9.3	EVENTSYSLOGSERVERCNF	89
4.10	FLASH MEMORY MANAGEMENT	90
4.10.1	GETFLASHMEMUSAGE	90
4.11	SYSTEM VOICEBOX NUMBER CONFIGURATION AND MAINTENANCE	90
4.11.1	GETSYSVOICEBOXNUM	90

4.11.2	SETSYSVOICEBOXNUM	90
4.11.3	EVENTSYSVOICEBOXNUMCNF	91
4.12	SYSTEM TONEScheme CONFIGURATION AND MAINTENANCE.....	91
4.12.1	TONESchemeTYPE	91
4.12.2	GETSYSTONEScheme	92
4.12.3	SETSYSTONEScheme.....	93
4.12.4	EVENTSYSTONESchemeCNF	93
4.13	AUTOMATIC DATABASE BACKUP AND RESTORE MANAGEMENT	93
4.13.1	GETAUTODDBACKUP	93
4.13.2	SETAUTODDBACKUP.....	94
4.13.3	GETAUTODDBACKUPFILENAME.....	94
4.13.4	EVENTAUTODDBACKUPCNF	94
4.13.5	EVENTAUTODDBACKUPFILENAMECNF	95
4.14	MANUAL DATABASE BACKUP AND RESTORE MANAGEMENT	95
4.14.1	MANUALDBBACKUP	95
4.14.2	MANUALDBRESTORE	95
4.14.3	DBBACKUPTOUSB	96
4.14.4	DBRESTOREFROMUSB.....	96
4.15	DATABASE BACKUP AND RESTORE STATE MANAGEMENT	96
4.15.1	GETDBTRANSFERSTATE.....	96
4.15.2	EVENTDBTRANSFERSTATE	97
4.16	TIME ZONE CONFIGURATION AND MAINTENANCE	98
4.16.1	TIMEZONELISTTYPE	98
4.16.2	TIMEZONEType.....	98
4.16.3	GETTIMEZONE	99
4.16.4	GETTIMEZONELIST.....	99
4.16.5	EVENTTIMEZONELIST	99
4.16.6	GETTIMEZONEDETAILS.....	100
4.16.7	SETTIMEZONE	100
4.16.8	SETTIMEZONEDETAILS	101
4.16.9	RESETTIMEZONEDETAILS.....	101
4.16.10	EVENTTIMEZONECNF	101
4.16.11	EVENTTIMEZONEDETAILSCNF.....	102

4.17	SNMP CONFIGURATION AND MAINTENANCE.....	102
4.17.1	GETSNMP.....	102
4.17.2	SETSNMP.....	102
4.17.3	EVENTSNMPCNF.....	103
4.18	REMOTE SYSTEM DUMP CONFIGURATION AND MAINTENANCE.....	103
4.18.1	ACTIVATEREMOTESYSTEMDUMP.....	103
4.18.2	GETREMOTESYSTEMDUMP.....	104
4.18.3	SETREMOTESYSTEMDUMP.....	104
4.18.4	EVENTREMOTESYSTEMDUMPCNF.....	105
4.19	HEALTH STATE MONITORING.....	105
4.19.1	GETHEALTHSTATE.....	105
4.19.2	EVENTHEALTHSTATE.....	106
4.20	EVENT LOG MONITORING.....	106
4.20.1	EVENTLOGENTRYTYPE.....	106
4.20.2	GETEVENTLOGBUFFER.....	106
4.20.3	DELETEEVENTLOGBUFFER.....	107
4.20.4	EVENTEVENTLOGENTRY.....	107
4.21	EXTERNAL USER DATA SERVER CONFIGURATION AND MAINTENANCE.....	108
4.21.1	GETUSERDATASERVER.....	108
4.21.2	SETUSERDATASERVER.....	108
4.21.3	EVENTUSERDATASERVERCNF.....	108
4.21.4	GETPRESERVEUSERDEVICERELATION.....	109
4.21.5	SETPRESERVEUSERDEVICERELATION.....	109
4.21.6	EVENTPRESERVEUSERDEVICERELATIONCNF.....	110
4.22	SYSTEM CONFIGURATION FILES AND MAINTENANCE FOR PROVISIONING.....	110
4.22.1	GETSYSTEMPROVUPDTRIG.....	110
4.22.2	SETSYSTEMPROVUPDTRIG.....	111
4.22.3	EVENTSYSTEMPROVUPDTRIGCNF.....	111
4.22.4	GETSYSTEMCREDENTIALS.....	111
4.22.5	SETSYSTEMCREDENTIALS.....	112
4.22.6	EVENTSYSTEMCREDENTIALSCNF.....	112
4.22.7	GETCONFIGURL.....	112
4.22.8	SETCONFIGURL.....	113

4.22.9	EVENTCONFIGURLCNF	113
4.22.10	GETUSEDCONFIGURL.....	113
4.22.11	EVENTUSEDCONFIGURL.....	113
4.23	USER DATA IMPORT AND MAINTENANCE.....	114
4.23.1	REQUESTMANUALUSERDATAIMPORT	114
4.23.2	EVENTUSERDATAIMPORT	114
4.24	NETWORK PARAMETERS CONFIGURATION AND MAINTENANCE	115
4.24.1	NETPARAMTYPE	115
4.24.2	GETNETPARAMS.....	115
4.24.3	SETNETPARAMS	115
4.24.4	EVENTNETPARAMSCNF	116
4.25	PARK / SARI CONFIGURATION AND MAINTENANCE	116
4.25.1	GETPARK	116
4.25.2	SETPARK.....	116
4.25.3	EVENTPARKCNF	117
4.25.4	PARKFROMSERVER	117
4.25.5	EVENTPARKFROMSERVERRESULT.....	117
4.25.6	GETSARI.....	118
4.25.7	SETSARI	118
4.25.8	EVENTSARICNF.....	118
4.26	DECT ENCRYPTION CONFIGURATION AND MAINTENANCE	119
4.26.1	GETDECTENCRYPTION.....	119
4.26.2	SETDECTENCRYPTION	119
4.26.3	EVENTDECTENCRYPTIONCNF.....	119
4.27	DECT REGULATORY DOMAIN CONFIGURATION AND MAINTENANCE	120
4.27.1	DECTREGDOMAINTYPE	120
4.27.2	GETDECTREGDOMAIN	120
4.27.3	SETDECTREGDOMAIN.....	120
4.27.4	EVENTDECTREGDOMAINCNF	121
4.28	DECT PAGING AREA SIZE CONFIGURATION AND MAINTENANCE	121
4.28.1	GETDECTPAGINGAREASIZE	121
4.28.2	SETDECTPAGINGAREASIZE.....	121
4.28.3	EVENTDECTPAGINGAREACNF	122

4.29	DECT AUTHENTICATION CODE CONFIGURATION AND MAINTENANCE	122
4.29.1	GETDECTAUTHCODE	122
4.29.2	SETDECTAUTHCODE	122
4.29.3	EVENTDECTAUTHCODECNF	123
4.30	DECT SUBSCRIPTION MODE CONFIGURATION AND MAINTENANCE	123
4.30.1	GETDECTSUBSCRIPTIONMODE	123
4.30.2	SETDECTSUBSCRIPTIONMODE	123
4.30.3	EVENTDECTSUBSCRIPTIONMODE	124
4.31	DECT PHONE SETTINGS CONFIGURATION AND MAINTENANCE	125
4.31.1	PPBOOTTEXTTYPE.....	125
4.31.2	GETDECTPPSETTINGS	125
4.31.3	SETDECTPPSETTINGS	125
4.31.4	EVENTDECTPPSETTINGSCNF	126
4.32	DECT POWER LIMIT CONFIGURATION AND MAINTENANCE.....	126
4.32.1	GETDECTPOWERLIMIT	126
4.32.2	SETDECTPOWERLIMIT.....	127
4.32.3	EVENTDECTPOWERLIMITCNF	127
4.33	AUTO-CREATE ON SUBSCRIPTION CONFIGURATION AND MAINTENANCE.....	127
4.33.1	GETDEVAUTOCREATE	127
4.33.2	SETDEVAUTOCREATE	127
4.33.3	EVENTDEVAUTOCREATECNF.....	128
4.34	SIP AND RTP SETTINGS CONFIGURATION AND MAINTENANCE	128
4.34.1	SIP CODECTYPE.....	128
4.34.2	SIPTRANSPORTPROTOCOLTYPE.....	128
4.34.3	SIPPORTRANGETYPE	129
4.34.4	GETBASICSIP	129
4.34.5	SETBASICSIP	130
4.34.6	EVENTBASICSIPCNF	131
4.34.7	GETADVANCEDSIP.....	132
4.34.8	SETADVANCEDSIP	133
4.34.9	EVENTADVANCEDSIPCNF.....	134
4.34.10	GETSECURESIP.....	136
4.34.11	SETSECURESIP	137

4.34.12	EVENTSECURESIPCNF.....	138
4.34.13	GETBACKUPSIP.....	139
4.34.14	SETBACKUPSIP.....	140
4.34.15	EVENTBACKUPSIPCNF.....	141
4.34.16	GETRTP.....	142
4.34.17	SETRTP.....	143
4.34.18	EVENTRTPCNF.....	144
4.34.19	GETDTMF.....	144
4.34.20	SETDTMF.....	145
4.34.21	EVENTDTMFCNF.....	145
4.34.22	GETREGISTRATIONTRAFFICSHAPING.....	146
4.34.23	SETREGISTRATIONTRAFFICSHAPING.....	146
4.34.24	EVENTREGISTRATIONTRAFFICSHAPINGCNF.....	147
4.34.25	GETSUPLSERV.....	147
4.34.26	SETSUPLSERV.....	148
4.34.27	EVENTSUPLSERVCNF.....	149
4.34.28	GETCONFERENCESERVERSIP.....	150
4.34.29	SETCONFERENCESERVERSIP.....	151
4.34.30	EVENTCONFERENCESERVERSIPCNF.....	151
4.34.31	GETPORTRANGESIP.....	152
4.34.32	SETPORTRANGESIP.....	152
4.34.33	EVENTPORTRANGESIPCNF.....	153
4.34.34	GETINTERCOMCALLHANDLINGSIP.....	153
4.34.35	SETINTERCOMCALLHANDLINGSIP.....	154
4.34.36	EVENTINTERCOMCALLHANDLINGSIPCNF.....	155
4.35	SECURE CERTIFICATE CONFIGURATION AND MAINTENANCE.....	156
4.35.1	SECURE PROVISIONING CERTIFICATE HANDLING.....	156
4.35.1.1	GetSecurePROVCertificate.....	156
4.35.1.2	SetSecurePROVCertificate.....	156
4.35.1.3	EventSecurePROVCertificateCnf.....	156
4.35.1.4	GetSecurePROVCertificateServerImport.....	156
4.35.1.5	SetSecurePROVCertificateServerImport.....	157
4.35.1.6	EventSecurePROVCertificateServerImportCnf.....	158
4.35.2	SECURE OMM CERTIFICATE HANDLING.....	158

4.35.2.1	GetSecureOMMCertificate	158
4.35.2.2	GetOMMCertificate	159
4.35.2.3	SetSecureOMMCertificate	159
4.35.2.4	SetOMMCertificate	160
4.35.2.5	EventOMMCertificateCnf	160
4.35.2.6	GetSecureOMMCertificateServerImport	161
4.35.2.7	SetSecureOMMCertificateServerImport	161
4.35.2.8	EventSecureOMMCertificateServerImportCnf	162
4.35.3	SECURE SIP CERTIFICATE HANDLING	162
4.35.3.1	GetSecureSIPCertificate	162
4.35.3.2	SetSecureSIPCertificate	163
4.35.3.3	EventSecureSIPCertificateCnf	164
4.35.3.4	GetSecureSIPCertificateServerImport	165
4.35.3.5	SetSecureSIPCertificateServerImport	165
4.35.3.6	EventSecureSIPCertificateServerImportCnf	166
4.36	DECT PHONE USER LOGIN ADMINISTRATION CONFIGURATION AND MAINTENANCE	166
4.36.1	GETPPLOGINVARIANT	166
4.36.2	SETPPLOGINVARIANT	167
4.36.3	EVENTPPLOGINVARIANTCNF	167
4.37	RESTRICTED SUBSCRIPTION DURATION CONFIGURATION AND MAINTENANCE	167
4.37.1	GETRESTRICTEDSUBSCRIPTIONDURATION	167
4.37.2	SETRESTRICTEDSUBSCRIPTIONDURATION	168
4.37.3	EVENTRESTRICTEDSUBSCRIPTIONDURATIONCNF	168
4.38	DECT PHONE DEVICE CONFIGURATION AND MAINTENANCE	169
4.38.1	PPDEVTYPE	169
4.38.2	GETPPDEVSUMMARY	171
4.38.3	GETPPDEV	171
4.38.4	CREATEPPDEV	172
4.38.5	DELETEPPDEV	172
4.38.6	SETPPDEV	173
4.38.7	EVENTPPDEVCNF	173
4.38.8	EVENTPPDEVSUMMARY	174
4.38.9	EVENTPPTRANSACTION	174
4.38.10	GETLASTPPDEVACTION	176
4.38.11	GETACTIVEPPDEV	177

4.39	DECT PHONE USER CONFIGURATION AND MAINTENANCE	177
4.39.1	PPUSERTYPE	177
4.39.2	GETPPUSERSUMMARY	182
4.39.3	GETPPUSER.....	182
4.39.4	CREATEPPUSER.....	183
4.39.5	CREATEFIXEDPP	184
4.39.6	DELETEPPUSER	184
4.39.7	SETPPUSER.....	185
4.39.8	EVENTPPUSERCNF.....	185
4.39.9	EVENTPPUSERSUMMARY.....	186
4.40	USER DEVICE RELATION CONFIGURATION AND MAINTENANCE.....	186
4.40.1	SETPPUSERDEVRELATION.....	186
4.41	TELEPHONY MONITORING	187
4.41.1	GETPPSTATE	187
4.41.2	EVENTPPSTATE.....	188
4.42	RFP CONFIGURATION AND MAINTENANCE.....	189
4.42.1	RFPTYPE	189
4.42.2	GETRFPSUMMARY.....	194
4.42.3	GETRFP.....	195
4.42.4	CREATERFP	196
4.42.5	DELETERFP	196
4.42.6	SETRFP	197
4.42.7	EVENTRFPCNF.....	197
4.42.8	EVENTRFPSTATE	198
4.42.9	EVENTRFPDETAILS.....	198
4.42.10	EVENTRFPCONNECTATTEMPT.....	199
4.42.11	EVENTRFPSUMMARY	199
4.42.12	CAPTUREDRFPTYPE.....	200
4.42.13	SETRFPCAPTURE.....	200
4.42.14	GETRFPCAPTURE	201
4.42.15	EVENTRFPCAPTURECNF	201
4.42.16	GETRFPCAPTURELIST	201
4.42.17	DELETERFPCAPTURELIST.....	202

4.42.18	DELETERFPCAPTURELISTELEM.....	202
4.42.19	EVENTRFPCAPTURE.....	203
4.43	RFPM CONFIGURATION AND MAINTENANCE.....	203
4.43.1	IPADDRESSTYPE.....	203
4.43.2	GETRFPM.....	203
4.43.3	SETRFPM.....	204
4.43.4	EVENTRFPMCNF.....	205
4.44	WLAN REGULATORY DOMAIN CONFIGURATION AND MAINTENANCE.....	205
4.44.1	GETWLANREGDOMAINLIST.....	205
4.44.1.1	WLANRegDomainType.....	205
4.44.2	GETWLANREGDOMAIN.....	206
4.44.3	SETWLANREGDOMAIN.....	206
4.44.4	EVENTWLANREGDOMAINCNF.....	207
4.45	WLAN PROFILE CONFIGURATION AND MAINTENANCE.....	207
4.45.1	SSIDPROFILETYPE.....	207
4.45.2	WLANPROFILETYPE.....	208
4.45.3	GETWLANPROFILE.....	209
4.45.4	CREATEWLANPROFILE.....	210
4.45.5	DELETEWLANPROFILE.....	210
4.45.6	SETWLANPROFILE.....	211
4.45.7	EVENTWLANPROFILECNF.....	211
4.46	WLAN ACCESS CONTROL LIST CONFIGURATION AND MAINTENANCE.....	211
4.46.1	ACLENTRYTYPE.....	211
4.46.2	CREATEACLENTRY.....	212
4.46.3	GETACLENTRY.....	212
4.46.4	SETACLENTRY.....	213
4.46.5	DELETEACLENTRY.....	214
4.46.6	EVENTACL CNF.....	214
4.47	WLAN CONNECTED CLIENTS MONITORING.....	215
4.47.1	WLANCLIENTTYPE.....	215
4.47.2	GETWLANCLIENTS.....	215
4.47.3	EVENTWLANCLIENT.....	216
4.48	PBX ENROLMENT MANAGEMENT.....	216

4.48.1	GETPBXENROLMENT	216
4.48.2	SETPBXENROLMENT.....	217
4.48.3	EVENTPBXENROLMENTCNF	217
4.49	SITE CONFIGURATION AND MAINTENANCE	218
4.49.1	SITETYPE	218
4.49.2	GETSITE SUMMARY.....	218
4.49.3	EVENTSITE SUMMARY	219
4.49.4	GETSITE	219
4.49.5	CREATE SITE	220
4.49.6	DELETE SITE.....	220
4.49.7	SET SITE	221
4.49.8	EVENTSITECNF	221
4.50	DIGIT TREATMENT CONFIGURATION AND MAINTENANCE	222
4.50.1	SITEIDTYPE	222
4.50.2	DIGIT TREATMENT TYPE	222
4.50.3	CREATEDIGIT TREATMENT	223
4.50.4	DELETEDIGIT TREATMENT	223
4.50.5	GETDIGIT TREATMENT.....	223
4.50.6	GETDIGIT TREATMENT SUMMARY	224
4.50.7	SETDIGIT TREATMENT	225
4.50.8	EVENTDIGIT TREATMENTCNF.....	225
4.51	FEATURE ACCESS CODES CONFIGURATION AND MAINTENANCE	225
4.51.1	FACTYPE	225
4.51.2	GETFACLIST	226
4.51.3	SETFAC	227
4.51.4	SETFACLIST.....	227
4.51.5	GETFACPREFIX.....	227
4.51.6	SETFACPREFIX	228
4.51.7	EVENTFACCNF.....	228
4.51.8	EVENTFACPREFIXCNF	229
4.52	CORPORATE DIRECTORY CONFIGURATION AND MAINTENANCE	229
4.52.1	LDAP CORPORATE DIRECTORIES	229
4.52.1.1	LDAPType	229

4.52.1.2 CreateLDAP	230
4.52.1.3 DeleteLDAP	230
4.52.1.4 GetLDAP	231
4.52.1.5 SetLDAP	231
4.52.1.6 EventLDAPCnf	231
4.52.2 XML CORPORATE DIRECTORIES	232
4.53 ALERTING CONFIGURATION AND MAINTENANCE	232
4.53.1 ALARMTRIGGERTYPE	232
4.53.2 GETALARMTRIGGERSUMMARY	233
4.53.3 GETALARMTRIGGER	233
4.53.4 CREATEALARMTRIGGER	234
4.53.5 DELETEALARMTRIGGER	234
4.53.6 SETALARMTRIGGER	235
4.53.7 EVENTALARMTRIGGERCNF	235
4.53.8 EVENTALARMTRIGGER	236
4.53.9 GENERATEHEALTHSTATEALARMTRIGGERS	239
4.53.10 EVENTALARMCALLPROGRESS	239
4.54 REMOTE SYSTEM DUMP CONFIGURATION AND MAINTENANCE	239
4.54.1 ACTIVATEREMOTESYSTEMDUMP	239
4.54.2 GETREOTESYSTEMDUMP	240
4.54.3 SETREOTESYSTEMDUMP	240
4.54.4 EVENTREOTESYSTEMDUMPCNF	241
4.55 MESSAGING	241
4.55.1 MESSAGE TYPE	241
4.55.2 MESSAGECONFIRMTYPE	249
4.55.3 SENDMESSAGE	249
4.55.4 DELETMESSAGE	250
4.55.5 CANCELMESSAGE	251
4.55.6 EVENTMESSAGEQUEUEEMPTY	252
4.55.7 EVENTMESSAGEPROGRESS	252
4.55.8 EVENTMESSAGECONFIRMATION	254
4.55.9 EVENTMESSAGESEND	254
4.56 INTEGRATED MESSAGE AND ALARM SERVER APPLICATION CONFIGURATION AND MAINTENANCE	254

4.56.1	GETIMA.....	254
4.56.2	SETIMA.....	255
4.56.3	EVENTIMACNF.....	255
4.57	OMP JAVA WEB START CONFIGURATION AND MAINTENANCE.....	256
4.57.1	GETOMPURL.....	256
4.57.2	SETOMPURL.....	256
4.57.3	EVENTOMPURLCNF.....	256
4.58	LOCATING CONFIGURATION AND MAINTENANCE.....	257
4.58.1	LOCATIONTYPE.....	257
4.58.2	VISIBILITYTYPE.....	257
4.58.3	REQUESTPOSITIONINFO.....	257
4.58.4	SETPPUSERTRACKING.....	258
4.58.5	EVENTPOSITIONHISTORY.....	258
4.58.6	EVENTPOSITIONINFO.....	259
4.58.7	EVENTPOSITIONTRACK.....	259
4.58.8	EVENTPOSITIONREQUEST.....	260
4.59	USER MONITORING CONFIGURATION AND MAINTENANCE.....	260
4.59.1	GETUSERMONITORING.....	260
4.59.2	SETUSERMONITORING.....	261
4.59.3	EVENTUSERMONITORINGCNF.....	262
4.60	XML APPLICATION CONFIGURATION AND MAINTENANCE.....	262
4.60.1	XMLAPPLICATIONTYPE.....	262
4.60.2	CREATEXMLAPPLICATION.....	264
4.60.3	DELETEXMLAPPLICATION.....	265
4.60.4	GETXMLAPPLICATION.....	265
4.60.5	SETXMLAPPLICATION.....	265
4.60.6	EVENTXMLAPPLICATIONCNF.....	266
4.61	CONFERENCE SERVER CONFIGURATION AND MAINTENANCE.....	266
4.61.1	CONFERENCEROOMTYPE.....	266
4.61.2	CONFERENCECODECTYPE.....	267
4.61.3	CONFERENCEDATATYPE.....	268
4.61.4	CREATECONFERENCEROOM.....	269
4.61.5	DELETECONFERENCEROOM.....	269

4.61.6	GETCONFERENCEROOM.....	270
4.61.7	SETCONFERENCEROOM	271
4.61.8	EVENTCONFERENCEROOMCNF.....	271
4.61.9	CREATECONFERENCE	271
4.61.10	DELETECONFERENCE	272
4.61.11	ADDUSERTOCONFERENCE	272
4.61.12	DELETEUSERFROMCONFERENCE	272
4.61.13	CHANGEUSERINCONFERENCE	273
4.61.14	SETRTPCONFERENCESTREAMCHG	273
4.61.15	EVENTCREATECONFERENCE	274
4.61.16	EVENTDELETECONFERENCE.....	274
4.61.17	EVENTADDUSERTOCONFERENCE	275
4.61.18	EVENTCHANGEUSERINCONFERENCE	275
4.61.19	EVENTDELETEUSERFROMCONFERENCE	275
4.61.20	EVENTRTPCONFERENCESTREAMCHG	276
4.61.21	EVENTCONFERENCEREQUEST	276
4.61.22	EVENTCONFERENCERELEASE.....	276
4.61.23	CONFERENCECONFIRMATION.....	277
4.61.24	GETG729CHANNELSFORCONFERENCE	277
4.61.25	GETREADYFORCONFERENCING	278
4.61.26	EVENTREADYFORCONFERENCING	278
4.61.27	GETFREECONFERENCECHANNELS	278
4.61.28	EVENTFREECONFERENCECHANNELS	279
4.62	BLUETOOTH BEACON CONFIGURATION AND MAINTENANCE	279
4.62.1	BTRFPSENSTYPE	279
4.62.2	BTPPSENSTYPE	279
4.62.3	BTBTYPE	280
4.62.4	GETBLUETOOTHGLOBALSETTINGS.....	281
4.62.5	SETBLUETOOTHGLOBALSETTINGS	281
4.62.6	EVENTBLUETOOTHGLOBALSETTINGS CNF.....	281
4.62.7	GETBLUETOOTHSENSITIVITY.....	282
4.62.8	SETBLUETOOTHSENSITIVITY	282
4.62.9	EVENTBLUETOOTHSENSITIVITYCNF.....	282

4.62.10	GETBLUETOOTHBEACON	283
4.62.11	CREATEBLUETOOTHBEACON	283
4.62.12	DELETEBLUETOOTHBEACON.....	284
4.62.13	SETBLUETOOTHBEACON.....	284
4.62.14	EVENTBLUETOOTHBEACONCNF	285
4.62.15	GETBLUETOOTHBEACONSUMMARY	285
4.62.16	EVENTBLUETOOTHBEACONSUMMARY	286
4.62.17	BTSTATISTICDATATYPE	286
4.62.18	BTCLIENTSTATISTICDATATYPE.....	287
4.62.19	GETBLUETOOTHCLIENTSTATISTIC.....	287
4.62.20	EVENTBLUETOOTHCLIENTSTATISTIC	288
4.63	VIDEO DEVICE CONFIGURATION AND MAINTENANCE	288
4.63.1	PPTYPE	288
4.63.2	VIDEODEVTYPE	288
4.63.3	GETVIDEODEV.....	289
4.63.4	CREATEVIDEODEV	290
4.63.5	DELETEVIDEODEV.....	290
4.63.6	SETVIDEODEV	291
4.63.7	EVENTVIDEODEVCNF	291
4.63.8	GETVIDEODEVSUMMARY	291
4.63.9	EVENTVIDEODEVSUMMARY	292
4.63.10	GETVIDEODEVLINK	292
4.63.11	EVENTVIDEODEVLINK.....	293
4.64	ADDITIONAL SETTINGS CONFIGURATION AND MAINTENANCE	293
4.64.1	GETADDITIONALSETTINGS.....	293
4.64.2	SETADDITIONALSETTINGS.....	294
4.64.3	EVENTADDITIONALSETTINGS CNF	294
4.65	SYNCHRONIZATION MONITORING	294
4.65.1	SYNCOFFSETTYPE	294
4.65.2	GETRFPSYNC	295
4.65.3	EVENTRFPSYNCREL.....	295
4.66	RFP QUALITY MONITORING	296
4.66.1	MEDIA STREAM QUALITY MONITORING.....	296

4.66.1.1 MsQualityType.....	296
4.66.1.2 GetRFPMediaStreamQuality	296
4.66.1.3 ResetRFPMediaStreamQuality	297
4.66.1.4 EventRFPMediaStreamQuality	297
4.66.2 IP QUALITY MONITORING	297
4.66.2.1 IpQualityType	298
4.66.2.2 GetRFPipQuality	298
4.66.2.3 EventRFPipQuality	299
4.66.3 SYNCHRONIZATION QUALITY MONITORING	299
4.66.3.1 SyncQualityType	299
4.66.3.2 GetRFPsyncQuality.....	299
4.66.3.3 EventRFPsyncQuality	300
4.67 DECT PHONE DEVICE PROFILE CONFIGURATION AND MAINTENANCE	300
4.67.1 PpPROFILEDATASETTYPE	301
4.67.2 GETPPPPROFILE.....	301
4.67.3 SETPPPPROFILE	302
4.67.4 EVENTPPPPROFILECNF	302
4.68 CORE DUMP CONFIGURATION AND MAINTENANCE	302
4.68.1 GETCOREDUMPURL	302
4.68.2 SETCOREDUMPURL	302
4.68.3 EVENTCOREDUMPURLCNF	303
4.69 NTP SERVER CONFIGURATION AND MAINTENANCE.....	303
4.69.1 GETNTPSERVER	303
4.69.2 SETNTPSERVER.....	303
4.69.3 EVENTNTPSERVERCNF	304
4.70 SPECIAL BRANDING CONFIGURATION AND MAINTENANCE.....	304
4.70.1 GETSPECIALBRANDING.....	304
4.70.2 SETSPECIALBRANDING	305
4.70.3 EVENTSPECIALBRANDINGCNF.....	305
4.71 USER DEVICE SYNCHRONIZATION CONFIGURATION AND MAINTENANCE	306
4.71.1 GETUSERDEVICESYNCOMM	306
4.71.2 SETUSERDEVICESYNCOMM	306
4.71.3 EVENTUSERDEVICESYNCOMM CNF	307
4.72 SYSTEM STATISTIC COUNTERS MANAGEMENT	308

4.72.1	SYSSTATNAMETYPE	309
4.72.2	GETSYSSTATISTICCONFIG.....	309
4.72.3	GETSYSSTATISTICOCCURRENCE	310
4.72.4	GETSYSSTATISTICMINMAXSUMMARY.....	310
4.72.5	SYSRECHEADERTYPE	311
4.72.6	SYSRECTYPE.....	311
4.72.7	GETSYSSTATISTICMINMAXRECORD.....	312
4.72.8	GETSYSSTATISTICMINMAX.....	312
4.72.9	RESETSYSSTATISTIC.....	313
4.73	RFP STATISTIC COUNTERS MANAGEMENT	314
4.73.1	RFPSTATHEADTYPE	315
4.73.2	RFPSTATNAMETYPE.....	315
4.73.3	GETRFPSTATISTICCONFIG.....	315
4.73.4	GETRFPSTATISTIC.....	316
4.73.5	RESETRFPSTATISTIC.....	316
4.74	FILE TRANSFER MANAGEMENT.....	317
4.74.1	GETFILE	317
4.74.2	PUTFILE.....	318
5	APPENDIX – OPENMOBILITY® TYPE DIFFERENCES.....	321
5.1	REQUESTS/EVENTS DEPENDENCIES OF “OPENRESP”-PARAMETERS	321
5.2	CONDITIONS FOR “OPENRESP”-PARAMETERS.....	327
5.3	OMM VERSION HISTORY	334
5.4	OPENMOBILITY® SIP.....	347
5.5	OPENMOBILITY® LICENSE CONDITIONS	348
5.6	EXAMPLES	349
5.6.1	OMP SYSTEM START UP.....	349
5.6.2	OMP STATISTIK START UP	367
5.7	OMM AXI XML SCHEMA.....	372

REVISION CONTROL INFORMATION

DATE	AUTHOR	VERSION	CHANGES
2012/10/31	Christian Krohne	4.2	<ul style="list-style-type: none"> • New <i>protocolVersion</i> (39). • New versions see 5.3 • Textual corrections • New event <i>EventConferenceRelease</i> and attribute changes in some conference requests/events • Change permission for <i>EventVideoDevLink</i> to Video • New Bluetooth attribute <i>failed</i> in <i>BTBType</i> • New values in <i>state</i> attribute of <i>EventAutoDBState</i> and <i>GetAutoDBState</i>
2013/08/21	Christian Krohne	5.1	<ul style="list-style-type: none"> • New <i>protocolVersion</i> (40). • New versions see 5.3 • Name changes Alarm trigger cause “AUTODDBTRANSFER” • New health state cause • New attribute <i>advancedFeatureErrorRFPs</i> <i>EventRFPSummary</i> and <i>GetRFPSummary</i> • Most attributes of <i>SetTzoneDetails</i> are optional • Support additional settings (4.64) / truncate long names in <i>CreatePPUser</i> and <i>SetPPUser</i> request and new attribute in <i>OpenResp</i> • Added comment for <i>SetDECTSubscriptionMode</i> (BugFix) • New attributes <i>hasEncryption</i> and <i>hasAdvancedFeatures</i> in <i>RFPTYPE</i> • New event <i>EventTimeZoneList</i> and new attribute in <i>GetTimeZoneList</i> • New attribute <i>videoStreaming</i>, <i>srtp</i> and <i>dectSecurity</i> in <i>SiteType</i> • New attribute <i>usedPagingAreas</i> in <i>GetRFPSummary</i> and <i>EventRFPSummary</i> • New attribute <i>sipRegisterCheck</i> in <i>PPUserType</i> • New attribute <i>sipRegisterCheckNum</i> in request/event <i>GetBasicSIP</i> and <i>EventBasicSIPConf</i> • New attribute in <i>EventDbTransferState</i> • <i>RFPType</i> list correction • <i>RFP quality monitoring</i> • PP / DECT phone configuration management added (HCM) • Secure SIP configuration added • License client/server configuration added • <i>LicenseType</i> changed and new <i>LicenseFile</i> subscription • New attributes <i>LimitsResp</i> and <i>OpenResp</i> • New <i>name</i> attribute in <i>ACLEntryType</i> and new request <i>SetACLEntry</i> • Changed video resolution settings and delete <i>rotation</i> attribute

DATE	AUTHOR	VERSION	CHANGES
			<ul style="list-style-type: none"> • New requests <i>GetBlueToothBeaconSummary</i>, <i>GetVideoDevSummary</i> and new events <i>EventBlueToothBeaconSummary</i>, <i>EventVideoDevSummary</i>. • New event types for subscriptions. • New attributes in <i>AddUserToConference</i> and <i>EventAddUserToConference</i> • New attributes <i>wlanChannelUsed</i>, <i>wlanPowerUsed</i> and <i>wlanHighThroughputTypeUsed</i> in <i>RFPTType</i> • New requests/event <i>BlueToothGlobalSettings</i>. • New requests/event <i>OMP</i>. • New Optional <i>SendMessage</i> attribute. • New requests/event <i>DECTPpSettings</i> • New attribute <i>callRejectStateCode</i> in <i>AdvancedSIP</i> settings and new attribute in <i>Open</i> response • Additional attributes in <i>GetSiteSummary</i> and <i>EventSiteSummary</i> • New request <i>UpdateIMAConfigFile</i>. • <i>EventRFPCCapture / GetRFPCCaptureList</i> Bugfix • <i>SoftwareUpdate</i> Bugfix
2013/09/12	Christian Krohne	5.2	<ul style="list-style-type: none"> • New <i>protocolVersion</i> (42). • Textual changes • Changed <i>state names</i> in <i>VideoDevType</i> • Changed attributes in <i>Get/SetDectPpSettings</i> and <i>EventDectPpSettingsCnf</i> • New attribute <i>nSipRegistration</i> in <i>GetPPUserSummary</i> and <i>EventPPUserSummary</i> • New requests and event for SIP certificate server import • Additional health states for SIP certificate server import
2013/11/06	Christian Krohne	5.3	<ul style="list-style-type: none"> • Textual changes • Obsolete attribute <i>method</i> in <i>SuplServ</i> element deleted • New attributes <i>callTransferByHook</i>, <i>uriSeparator</i> and <i>reRegisterAfterFailOver</i> in <i>SuplServ</i> configuration • New <i>Limits</i> attribute <i>wlanProfiles</i>
2014/11/27	Christian Krohne	6.0	<ul style="list-style-type: none"> • New <i>protocolVersion</i> (43). • PP / DECT phone configuration management (HCM) rework • New attributes <i>releaseInfoTimerActiveCall</i>, <i>releaseInfoTimerFailedCall</i>, <i>releaseInfoTimerHoldCall</i>, <i>ringingOnHold</i> and <i>transferByHook6xxd</i> in <i>SuplServ</i> configuration • <i>batteryLevel</i> in <i>GetPPState</i> and <i>EventPPState</i> depends on user monitoring activation • New requests/event for <i>RestrictedSubscriptionDuration</i> • Additional attributes <i>authenticationTLS</i> and <i>commonNameValidationTLS</i> in <i>Get/SetSecureSIP</i>

DATE	AUTHOR	VERSION	CHANGES
			<p>and <i>EventSecureSIPCnf</i></p> <ul style="list-style-type: none"> • New attributes in <i>PPUserType</i> for user credentials (to user configuration files on external server) and DECT phone configuration data. • New requests <i>Get/SetSystemCredentials</i>, <i>Get/SetConfigURL</i>, <i>Get/SetUsedConfigURL</i> and new events <i>EventSystemCredentialsCnf</i>, <i>EventUsedConfigURL</i> and <i>EventConfigURLCnf</i>. Additional <i>URLType</i> changes. • New element <i>SystemCredentialPasswd</i> in <i>FeatureType</i> to support system credential password configuration by using the DECT phone. • Element <i>PINChange</i> in <i>FeatureType</i> deleted. • New requests <i>Get/SetSpecialBranding</i> and new event <i>EventSpecialBrandingCnf</i>. New attribute <i>haveSpecialBranding</i> in <i>Open</i> • Additional return code “<i>ELicenseWrongInstallId</i>” • Remove <i>Get/SetAutoDBRestore</i> and <i>EventAutoDBRestoreCnf</i> • <i>DeletePpUser</i> accepts also the number to delete the user • SIP port ranges added: New attributes <i>fixedSipPort</i> and <i>calculatedSipPort</i> in <i>PPUserType</i> and <i>ConferenceRoomType</i> New <i>Get/SetPortRangeSIP</i> and new event <i>EventPortRangeSIPCnf</i> • New IMA configuration • New file down- and upload for IMA configuration file in <i>PutFile</i> and <i>GetFile</i> • SIP intercom call handling added: New attributes <i>autoAnswer</i>, <i>microphoneMute</i>, <i>warningTone</i> and <i>allowBargeln</i> in <i>PPUserType</i>. New <i>Get/SetIntercomCallHandlingSIP</i> and new event <i>EventIntercomCallHandlingCnf</i> • New health state for AXI command processing in provisioning files and provisioning server. • System provisioning update trigger added: New <i>Get/SetSystemProvUpdTrig</i> and new event <i>EventProvUpdTrigCnf</i> • PARK service from PARK server added • New attribute <i>callRejectStateCodeDev</i>, <i>sessionTimer</i>, <i>incomingCallTimeout</i> and changed attribute <i>callRejectStateCode</i> to <i>callRejectStateCodeUsr</i> in <i>Get/SetAdvancedSIP</i> and <i>EventAdvancedSIPCnf</i> • New requests <i>Get/SetCoreDumpURL</i>, <i>Get/SetSoftwareImageURL</i>, <i>Get/SetPPFirmwareURL</i>, and new events <i>EventCoreDumpURLCnf</i>, <i>EventSoftwareImageURLCnf</i>, <i>EventPPFirmwareURLCnf</i>.. • New requests <i>Get/SetNTPServer</i>, and new event <i>EventNTPServerCnf</i> and additional new attribute

DATE	AUTHOR	VERSION	CHANGES
			<p><i>haveNTP</i> in <i>OpenResp</i>.</p> <ul style="list-style-type: none"> • Changed requests/event <i>OMP</i> → <i>OMPURL</i>. • Rework SIP element settings. • New requests <i>Get/SetOMMCertificate</i> and event <i>EventOMMCertificateCnf</i>. • New optional attribute <i>forward</i> in <i>EventSyslogServerCnf</i> and <i>GetSyslogServer</i>, <i>SetSyslogServer</i>. • New requests <i>Get/SetPreserveUserDeviceRelation</i> and new event <i>EventSetPreserveUserDeviceRelationCnf</i>. • New attribute <i>haveRFPOMM</i> in <i>OpenResp</i> and new attribute <i>resetToFactoryDefaults</i> in <i>SystemRestart</i>, • New attribute <i>haveSoftwareImageURL</i> in <i>OpenResp</i>. <i>Get/SetSoftwareImageURL</i> and <i>EventSoftwareImageURLCnf</i> depend on <i>haveSoftwareImageURL</i>
2015/03/13	Christian Krohne	6.0.2	<ul style="list-style-type: none"> • New <i>protocolVersion</i> (44). • Textual corrections and Mitel Branding • New events/requests: <i>EventSecureOMMCertificateServerImportCnf</i>, <i>EventSecurePROVCertificateServerImportCnf</i>, <i>GetSecureOMMCertificate</i>, <i>GetSecureOMMCertificateServerImport</i>, <i>GetSecurePROVCertificateServerImport</i>, <i>SetSecureOMMCertificate</i>, <i>SetSecureOMMCertificateServerImport</i>, and <i>SetSecurePROVCertificateServerImport</i> • New <i>RegistrationTrafficShaping</i> parameters <i>renewalTimer</i> and <i>spreadRegRenew</i> • New attribute <i>ommStbRunning</i> in <i>RFPTType</i> • Remove RFPM attributes from <i>RFPTType</i> • New requests/event <i>EventRFPMCnf</i>, <i>GetRFPM</i> and <i>SetRFPM</i> • New requests/event <i>EventRemoteSystemDumpCnf</i>, <i>Get/SetRemoteSystemDump</i> and <i>ActivateRemoteSystemDump</i> • Use common certification parameter in URL for "SecureOMMCertificateServerImport", "SecurePROVCertificateServerImport" and "SecureSIPCertificateServerImport" • New health state for DHCP server • Additional return code "ELicenseFile"
2015/10/21	Christian Krohne	6.1	<ul style="list-style-type: none"> • Textual changes and bugfixes • New events/requests: <i>EventUserDeviceSyncOMMCnf</i>, <i>GetUserDeviceSyncOMM</i>, <i>SetUserDeviceSyncOMM</i>, • New attributes "haveUserDeviceSync" and "haveSARI" in <i>OpenResp</i> • New subscription "Roaming" • New attribute "timeStamp" in "PPUserType",

DATE	AUTHOR	VERSION	CHANGES
			<p>“PPDevType”, “Ping”, “PingResp” and “PpProfileDataSetType”</p> <ul style="list-style-type: none"> • Changed health states • New events/requests: <i>EventSARICnf</i>, <i>GetSARI</i>, <i>SetSARI</i>, • New attribute “<i>subscribeToPARIOnly</i>” in <i>PPDevType</i>. • New events/requests: <i>EventDECTPowerLimitCnf</i>, <i>GetDECTPowerLimit</i>, <i>SetDECTPowerLimit</i> • New regulatory domain value “<i>Radio1910_1927MHz250mW</i>” • New New attribute value “<i>ExternalBlindTransfer</i>” of “<i>conferenceServerType</i>” in “<i>Get/SetConferenceServerSIP</i>”, “<i>EventConferenceServerSIPCnf</i>”, “<i>CreateFixedPP</i>”, “<i>CreatePPUser</i>”, “<i>Get/SetPPUser</i>” and “<i>EventPPUserCnf</i>” • New feature access code for “BlindTransfer” • New attribute “<i>userAgentCompatibility</i>” in “<i>Get/SetAdvancedSIP</i>” and “<i>EventAdvancedSIPCnf</i>” • New XML applicatipon hooks • Add <i>Monitor</i> permission to statistic and quality requests • Remove attribute <i>haveLDAPSimple</i> in <i>OpenResp</i>

REFERENCED DOCUMENTS

REFERENCE TITLE

DEFINITIONS AND ABBREVIATIONS

ACRONYM	EXPLANATION
DECT	DIGITAL ENHANCED CORDLESS TELECOMMUNICATIONS
FAC	FEATURE ACCESS CODE
IMA	INTEGRATED MESSAGE AND ALARM SERVER APPLICATION
IPEI	INTERNATIONAL PORTABLE EQUIPMENT IDENTITY, UNIQUE ID FOR EACH DECT PHONE HARDWARE
L-RFP	HARDWARE LICENSED RFP
MWI	MESSAGE WAITING INDICATION
OM AXI	OPENMOBILITY® APPLICATION XML INTERFACE OM APPLICATION XML INTERFACE
OMM	OPENMOBILITY® MANAGER
OMP	OPENMOBILITY® MANAGER PORTAL
PARI	PRIMARY ACCESS RIGHT IDENTITY
PARK	PORTABLE ACCESS RIGHTS KEY
PP	PORTABLE PART, A DECT PHONE
RCS	REMOTE CONFIGURATION SERVER
RFP	RADIO FIXED PART, A DECT BASE STATION
SARI	SECONDARY ACCESS RIGHT IDENTITY
SIP	SESSION INITIATION PROTOCOL
SSL	SECURITY SOCKET LAYER - TRANSPORT LAYER SECURITY
TCP	TRANSPORT CONTROL PROTOCOL
UAK	USER AUTHENTICATION KEY, USED BETWEEN RFP AND PP
URL	UNIFORM RESOURCE IDENTIFIER
URI	UNIFORM RESOURCE LOCATOR
XML	EXTENSIBLE MARKUP LANGUAGE

1 Introduction

1.1 Summary

The OM Application XML Interface (OM AXI) is an application programming interface which allows 3rd party software to configure the OpenMobility® Manager (OMM) and to access several of its application interfaces.

The interface is designed not to be bound to a certain OMM version. To eliminate hardware dependency and to simplify the specification and implementation on, client and server side, an XML based language is used.

The communication takes place over an SSL link, where the OMM acts as server. Additionally it is possible to use unencrypted TCP.

2 Protocol Mechanisms

2.1 Connection States

A connection between a client and the OM AXI server, which runs in the OMM, can have several states. These are listed below

- *New*: The transport layer has been set up, but the authentication did not take place yet.
- *Active*: The authentication was successful.

Refer to chapter 2.2 for the explanation when there are transitions between these states and which actions are allowed in each state.

2.2 Session Management

The OMM sets up a listening SSL socket on port 12622 and a plain TCP socket on port 12621 and waits for incoming connections. It is not recommended to use unencrypted sockets, because the data stream may contain sensitive information. Both kinds of connections are referred to by the term *TCP* below. For encrypted sockets, the OMM supports TLSv1.0, TLSv1.1 and TLSv1.2.

After a client set up a TCP connection to OM AXI, the session is considered as *new*. In this state only the *Open* request (chapter 4.1) is accepted by OM AXI.

The client must send an *Open* request to the OMM containing certain setup information first. If the OM AXI cannot accept these settings, it sends a negative response; the client may send additional *Open* requests with updated ingredients to successfully set up a session.

If the client is not able to set up a valid session 30 seconds after connecting, the OMM closes the TCP link. It does also close the link immediately when it receives an unexpected, unsupported or malformed request from the client.

After the authentication succeeded, the link is in the state *active*. In this state all other requests are allowed to be sent by the client.

The session remains *active* as long as the TCP link is connected or until the client has been inactive for a while. A disconnection by whatever reason will automatically end the session.

If a client did not send a request for more than 5 minutes it is considered as inactive and the OMM will close the link. Events sent by the OMM to the client do not extend this timer. The request *Ping* (see 4.1.5) can be used by the client to extend the time.

To avoid heavy system load caused in opening an SSL connection and connecting to OM AXI with an *OpenRequest/Resp* the client should not use a new connection for each request. Several requests should be used within an OM AXI session.

2.3 Message Transport And Encoding

All requests and responses are sent over the TCP link. As TCP is a stream without any message separation, each message is terminated by one zero byte 0x00.

There is no special limit on the message sizes, but for keeping the system responsive, client and server must not send or ask for messages being larger than 64 Kbytes.

A client may decide to send only one request at once and wait for the response before continuing. But for performance reasons it may be favorable to send additional requests before receiving a response for an earlier one. For this procedure all requests and responses may have sequence numbers. These do not need to be consecutive but may be any values useful for the client.

Even if all messages have an XML format, no XML headers must be sent. All messages are UTF-8 encoded, which is compatible to US-ASCII if only that subset of characters is used. XML namespace declarations must not be used. Any special characters must be encoded as defined in the XML Version 1.0 standard.

The message types declared in this document mostly use XML attributes instead of elements, because this keeps the overhead smaller. The descriptions use the term *field* for both, attributes and elements. To find out how they are actually defined, refer to the XML Schema in chapter 5.6.

2.4 Encryption

Some attributes containing private information of users, e. g. passwords, are encrypted asymmetrically. To enable clients being implemented in a wide range of programming languages and to use common libraries, the standard RSA algorithm according to PKCS #1 v1.5 is used. The key length is 512 bits, which is a good compromise of high security requirements and low message overhead.

The client can obtain the public key using the request *GetPublicKey* (see 4.1.9).

When this algorithm and padding are used, up to 53 characters can be encrypted in one block ($512 / 8 - 11$). If the attribute is longer, it is encrypted block-wise. The encrypted blocks are concatenated which results in a binary string which has a length of $n * 64$ bytes ($n * 512 / 8$).

This binary string is converted to text using Base64 encoding according to RFC 4648. The normal Base 64 alphabet is used which contains the characters '+' and '/' with no line breaks.

The affected attributes are always encrypted using the public key, no matter if they are sent from or to OM AXI.

2.5 Permission

When *Open* is accepted by OM AXI, the response contains a list of permissions for that session. These permissions depend on the user account but may also have other factors which influence the permissions assigned. Therefore the client should not make any assumptions about the permissions assigned to a certain user.

The initial user accounts and passwords are identical with the accounts used by the Web Service Interface. However, additional user accounts may be set up which are only available on OM AXI.

2.6 General Rules

Messages defined in this document may contain optional and mandatory fields (e. g. XML attributes). Fields may be empty or may have a value.

If a field is present but has an empty value, this has the explicit meaning that the sender transmits an empty value. If a field is not part of a message, it means that the sender does not send the value. The latter case does not mean that the value is empty.

3 Common Data Types

The protocol uses some basic types which are described in this chapter.

3.1 Requests

The attributes and elements defined here are common to all requests.

Name	Type	Mandatory	Description
seq	Integer	no	The client may give a unique sequence number to each request it sends. The response to the request will contain the same sequence number. This can be used by the client to associate responses to the corresponding requests.

The sequence number (*seq*) is returned by OM AXI transparently in the response. The client may use any kind of cookie which is useful for it.

3.2 Responses

The attributes and elements defined here are common to all responses.

Name	Type	Mandatory	Description
seq	Integer	no	Sequence number taken from the original request. Only filled in if the original request also had a sequence number.
errCode	enumeration	no	Fixed string codes described in the documentation for the particular requests. If this attribute is absent it implies that no error occurred.
info	String	no	This field may contain additional information about an error. It is free style text in English and used mainly to track errors.
bad	String	no	In case of <i>EInval</i> , <i>EMissing</i> or <i>ETooLong</i> : Name of the invalid attribute or element.
maxLen	Integer	no	In case of <i>ETooLong</i> : Maximal length (in characters, not bytes) expected for the erroneous attribute.

3.3 Error Codes

These are possible values for *errCode*.

Value	Description
EAreaFull	Paging area is full
EAuth	Authentication failed, e. g. user name or password may be wrong
EDectRegDomainInvalid	A DECT regulatory domain must be set before the request can be performed
EEncryptNotAllowed	DECT encryption cannot be activated because of at least one RFP not supporting encryption is connected.
EExist	A field describing a unique attribute does already exist

EFailed	Request could not be fulfilled
EForbidden	This operation is not permitted with that instance
EInProgress	Another transaction is in progress, this request cannot be fulfilled currently
EInval	A field contains invalid data or exceeds a limit
EInvalidChars	A string contains invalid characters.
ELicense	Operation cannot be fulfilled because of license restrictions
ELicenseFile	Operation cannot be fulfilled because of license file restrictions
ELicenseWrongInstallId	Operation cannot be fulfilled because of an invalid installation id in the license file
EMissing	A field which is mandatory on this OMM version is missing
ENoEnt	No record found for given key or id
ENoMem	No more data sets can be created
EPerm	No sufficient permissions for this request
EPwEmpty	A required password was not specified.
EPwSimilarToHost	A new given password is too similar to the host name.
EPwSimilarToName	A new given password is too similar to the user name.
EPwTooManySimilarChars	A new given password too many similar characters.
EPwTooShort	A new given password is too short.
EPwTooSimilar	A new given password is too similar to the previous one.
EPwTooWeak	A new given password is too weak.
EPwUnchanged	A new given password does not differ to the old one.
ETooLong	A string is too long
EWlanRegDomainInvalid	A DECT regulatory domain must be set before the request can be performed

Which of these error codes actually can occur in certain *Responses* is not describe in detail. But if the meaning of the above error codes differs in certain context *Responses*, a specific description for the related error codes is added in the corresponding chapters.

3.4 ProtocolType

This ProtocolType contains types of transfer protocols.

These are possible values for ProtocolType type:

Value	Description
FTP	FTP protocol
FTPS	FTPS protocol
SFTP	SFTP protocol
HTTP	HTTP protocol
HTTPS	HTTPS protocol
TFTP	TFTP protocol
SCP	SCP protocol
None	No protocol

3.4.1 PEMCertificateType

The *PemCertificateType* type contains the following entries.

Name	Type	Description
key	String	Base64 encoded portion data of a PEM file. This includes the “Begin” and “End” parts of the certificate.

3.4.2 PEMCertificateListType

Name	Type	Description
certificate	sequence of PemCertificateType	List of all certificates within a PEM file. This list might be empty which means a deletion in a set request.

3.5 URLType

This type contains all fields of an URL. It is used in different requests and responses defined in this document.

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, this protocol is enabled
protocol	ProtocolType	no	Type of the used protocol to the server
host	string	no	Server name or address
port	integer	no	Server port number. This value is 0, when the default protocol port number (<i>ProtocolType</i>) is used.
path	string	no	Server directory and/or filename
username	string	no	Optional user name for server access

password	string	no	Optional password for server access, encrypted with public key
sslMethod	string	no	SSL version to be used for SSL server connection. One of “TLS1.0”, “TLS1.1”, “TLS1.2” or “Auto” for automatic negotiation between TLS 1.0 - 1.2
useCommonCerts	boolean	no	„1” or “true”, if the certificate settings of the configuration file URL (see 4.21.4 ff) for provisioning has to be used to access the server.

Certificate parameters:

privateKeyPassword	string	no	Optional password for the private key certificate access, encrypted with public key. If not set this string is empty!
validateCerts	boolean	no	„1” or “true”, the server certificate will be validated against trusted CA's (signed by a CA from the Mozilla CA certificate list). If trusted self-signed certificates are imported to the OMM, this parameter needs to be set to 0.
validateExpires	boolean	no	„1” or “true”, if the client verifies whether or not a certificate has expired prior to accepting the certificate.
validateHostName	boolean	no	„1” or “true”, if the validation of host names on the OMM is enabled.
allowNonConfTrustCerts	boolean	no	„1” or “true”, if .this parameter disables any server certificate validation as long as no trusted certificate was imported into the OMM. AXI commands in a received configuration file may import such trusted certificate into the OMM. This parameter will be ignored, if <i>validateCerts</i> is set to 1.
importCerts	boolean	no	„1” or “true”, if in combination with <i>validateCerts</i> also set to 1, the trusted certificate will be imported from the cert chain delivered in server response without any validation, as long as no trusted certificate was imported previously into the OMM.
nTrustedCertificates	integer	no	Number of set trusted certificates. This value is read only.
nLocalCertificates	integer	no	Number of set local certificate chains. This value is read only.
nPrivateKeys	integer	no	Number of set private keys. This value is read only.
trustedCertificates	PemCertificateList Type	no	All trusted certificates to set. Before setting the new trusted certificates in the OMM all old trusted certificate settings will be deleted. If this element is missing, the trusted certificates stay unchanged. This value is write only.

localCertificates	PemCertificateList Type	no	The local certificate chain to set. Before setting the new local certificate chains in the OMM all old local certificate chain settings will be deleted. If this element is missing, the local certificate chain stay unchanged. This value is write only.
privateKeys	PemCertificateList Type	no	One OMM private key to set. This key is used in conjunction with the local certificate chain to enable a connected client to verify the server. If this element is missing, the private key stays unchanged. This value is write only.

The common URLType is used different. The usage for each used URL is as follows:

AutoDBBackup URL:

<i>enable:</i>	used
<i>useCommonCerts</i>	used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used

CertificateServerImport URLs;

<i>enable:</i>	used
<i>useCommonCerts</i>	used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used

Config URL:

<i>enable:</i>	used
<i>useCommonCerts</i>	not used
<i>username and password</i>	used
certificate attributes:	used
certificate <i>privateKeyPassword</i>	used

CoreDump URL:

<i>enable:</i>	used
<i>useCommonCerts</i>	not used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used

IMA URL:

<i>enable:</i>	used
<i>useCommonCerts</i>	used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used

ManualDBBackup URL:

<i>enable:</i>	not used
<i>useCommonCerts</i>	used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used

ManualDBRestore URL:

<i>enable:</i>	not used
<i>useCommonCerts</i>	used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used

OMP URL:

<i>enable:</i>	used
<i>useCommonCerts</i>	not used

<i>username and password</i>	not used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used
PBXEnrolment URL:	
<i>enable:</i>	not used
<i>useCommonCerts</i>	not used
<i>username and password</i>	not used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used
PPFirmware URL:	
<i>enable:</i>	used
<i>useCommonCerts</i>	used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used
RemoteSystemDump URL:	
<i>enable:</i>	used
<i>useCommonCerts</i>	used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used
SoftwareImage URL:	
<i>enable:</i>	used
<i>useCommonCerts</i>	used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used
SpecialBranding URL:	
<i>enable:</i>	used
<i>useCommonCerts</i>	used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used
UserDataServer URL:	
<i>enable:</i>	used
<i>useCommonCerts</i>	used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used
XMLApplication URLs:	
<i>enable:</i>	not used
<i>useCommonCerts</i>	not used
<i>username and password</i>	used
certificate attributes:	not used
certificate <i>privateKeyPassword</i>	not used

3.6 PermissionType

PermissionType is an enumeration value defined as follows:

Value	Description
AllCnfWrite	Permission to write to the configuration database, will automatically set permission <i>AllCnfRead</i>
AllCnfRead	Permission to read all data from configuration database
InfoMessaging	Permission to send messages only for priority <i>Info</i> and to subscribe to message events

Messaging	Permission to send messages and to subscribe to message events, except with priority <i>Emergency</i> and <i>LocatingAlert</i> ; will automatically set permission <i>InfoMessaging</i> .
Alerting	Permission to send messages with priority <i>Emergency</i> , will automatically set permission <i>Messaging</i> and <i>InfoMessaging</i>
LocatingAlert	Permission to send messages with priority <i>LocatingAlert</i> , will automatically set permission <i>Messaging</i> and <i>InfoMessaging</i>
Locating	Permission to query the position of DECT phones and to track DECT phone positions
Monitoring	Permission to monitor various technical aspects of the mobility system, will automatically set permission <i>AllCnfRead</i>
Conferencing	Permission to use conferencing in the open mobility system will automatically set permission <i>AllCnfRead</i> .
Video	Permission to use video in the mobility system, will automatically set permission <i>AllCnfRead</i> .

3.7 DateTimeType

This type contains all fields of a date and time value. It is used in different requests and responses defined in this document.

Name	Type	Mandatory	Description
Year	integer	yes	Year, value: e. g. 2010
Month	integer	yes	Month, value: 1 >= month >=12
Day	integer	yes	Day, value: 1 >= day >=31
Hour	integer	yes	Hour, value: 0 >= hour >=23
Minute	integer	yes	Minute, value: 0 >= minute >=59
Second	integer	no	Second, value: 0 >= second >=59

3.8 HealthType

This type contains all fields of a Health value. It is used in different requests and responses defined in this document.

Name	Type	Mandatory	Description
component	ComponentType	yes	Identifies the health component
severity	SeverityType	yes	The severity of the component
reason	string	no	Additional reason for the component
reasonCode	integer	no	Additional value specifying the specific health state

ComponentType can have one of the following values:

Value	Description
sync	Health state of RFP synchronization
standby	Health state of standby OMM mechanism
dbTransfer	Health state of database import and export.
download	Health state of software download over air
rfp	Health state of RFP
license	Health state of licensing
ima	Health state of IMA
licenseG729	Health state of G729 licensing
umo	Health state of user monitoring
uds	Health state of user data server import Note: The state of the user data server will only be updated each time when the update interval of the common user data file elapsed.
usbMem	Health state of the USB memory
licenseServer	Health state of the license server
video	Health state of the video cameras
bluetooth	Health state of the Bluetooth beacons
sip	Health state of the SIP stack
sipCert	Health state of SIP certificate server import
axiProvisioningCommands	Health state of provisioning file processing concerning AXI commands
provisioningServer	Health state of provisioning server access
dhcpServer	Health state of DHCP server configuration
userDeviceSync	Health state of user/device synchronization

SeverityType can have one of the following values:

Value	Description
OK	State okay
Idle	This component is not active or used
Warning	There are problems, but the system is still working

Error	There are problems which make the system or essential parts of it fail
-------	--

ReasonCode values can have one of the following reason meanings:

ReasonCode	Reason
0	OK: No error.
1	<i>Warning license with no redundancy:</i> One license RFP is not connected.
26	<i>Warning all channels in use:</i> G.729 license forbids usage of further channels.
2	<i>Error blocked license:</i> License is violated by configuration.
3	<i>Error inactive license:</i> Insufficient number of license RFPs is connected.
4	<i>Error no license:</i> License is missing.
25	<i>Error invalid license period ended:</i> The period working with a violated license ended.
5	<i>Warning RFP synchronisation warning split cluster:</i> The synchronisation of a cluster is lost, all RFPs of the cluster will be resynchronised.
6	<i>Warning standby connection broken:</i> The connection to the standby OMM is disconnected.
7	<i>Warning manual database import:</i> A manual database import failed, see syslog for more details.
9	<i>Warning manual database export:</i> A manual database export failed, see syslog for more details.
10	<i>Warning automatic database export:</i> An automatic database export failed, see syslog for more details.
11	<i>Warning download over air file not available or corrupted:</i> The DECT phone software cannot be updated, because of a wrong or missing DECT phone software file.
12	<i>Warning RFP branding mismatch:</i> RFP(s) with wrong branding detected.
24	<i>Warning RFP standby mismatch:</i> RFP(s) with wrong OMM standby configuration.
13	<i>Warning RFP no encryption:</i> RFP(s) not supporting DECT encryption detected.
14	<i>Warning RFP protocol mismatch:</i> RFP(s) with not supported SW version detected.
30	<i>Warning RFP USB overload detected.</i>
37	<i>Warning RFP advanced features:</i> An RFP without advanced features is connected to a site, where advanced features are activated.
15	<i>IMA connection to OMI:</i> Connection cannot be established or connection lost.
16	<i>IMA read confirmation:</i> IMA configuration file cannot be received.
17	<i>IMA message queue full:</i> High message queue level.

18	<i>IMA send message:</i> Messages cannot be sent.
19	<i>IMA delete message:</i> Messages cannot be deleted.
20	<i>IMA send mail:</i> Error during email send occurred.
21	<i>IMA receive mail:</i> Error during email receives occurred.
22	<i>IMA delete mail:</i> Error during email delete occurred.
23	<i>IMA get RSS feed:</i> Error during RSS feed receive occurred.
27	<i>Warning user monitoring is in start up phase.</i> Activated user Monitoring starts running after start up phase.
28	<i>Warning too many users for user monitoring configured.</i> User Monitoring may cause a too heavy system load due to too many monitored users are configured.
29	<i>Error user monitoring loses OMM AXI connection.</i> User Monitoring does not work correctly, because of an interrupted OMM AXI connection
31	<i>Error user data server:</i> Import of the common user data file from server failed
32	<i>Error user data server:</i> Invalid configuration data in common user data file
33	<i>Warning user data server:</i> Import of at least one user data file from server failed
34	<i>Warning user data server:</i> Invalid configuration data in at least one user data file
35	<i>Warning USB memory:</i> USB memory is missing
36	<i>Warning download over air is in start up phase:</i> Activated DECT phone software file downloading starts running after start up phase.
40	<i>Error License Server:</i> Configured server not started.
41	<i>Error License Server:</i> License client not connected.
42	<i>Error License Server:</i> License file parse error.
43	<i>Error License Server:</i> Received no license from server.
44	<i>Error / Warning License Server:</i> Missing standby license server(s).
45	<i>Error License Server:</i> The configured installation ID does not fit to the license file.
46	<i>Error Video:</i> An error of a video device was detected
47	<i>Error Video:</i> A plugged and activated video device gets unplugged
48	<i>Error Bluetooth beacon:</i> An error of a Bluetooth beacon was detected
49	<i>Error Bluetooth beacon:</i> At least one Bluetooth beacon is unplugged

50	<i>Error SIP certificate import: HTTP error for SIP secure trusted certificate</i>
51	<i>Error SIP certificate import: Imported SIP secure trusted certificate not stored in database</i>
52	<i>Error SIP certificate import: File transfer error (wrong configuration?) for SIP secure trusted certificate</i>
53	<i>Error SIP certificate import: HTTP error for SIP secure local certificate</i>
54	<i>Error SIP certificate import: Imported SIP secure local certificate not stored in database</i>
55	<i>Error SIP certificate import: File transfer error (wrong configuration?) for SIP secure local certificate</i>
56	<i>Error SIP private key import: HTTP error for SIP secure private key</i>
57	<i>Error SIP private key import: Imported SIP secure private key not stored in database</i>
58	<i>Error SIP private key import: File transfer error (wrong configuration?) for SIP secure private key</i>
59	<i>Error AXI command processing: AXI command error(s) during provisioning file processing</i>
60	<i>Error provisioning server: Access to provisioning server failed</i>
61	<i>Error dhcpServer: Invalid IP address range configuration in DHCP server (range too small for number of RFPs)</i>
62	<i>Error user/Device synchronization: Cannot connect to external AXI / lost AXI connection</i>
63	<i>Error user/Device synchronization: Different AXI or OMM version</i>
64	<i>Error user/Device synchronization: Connection between RFP and PC OMM</i>
65	<i>Warning user/Device synchronization: Different system times</i>
66	<i>Error user/Device synchronization: User/device synchronization is activated on configured central OMM, too</i>
67	<i>Error user/Device synchronization: SARI not configured on central OMM</i>

3.9 PPUseCallStateType

This type describes the call state of a DECT phone

PPUseCallStateType is an enumeration value defined as follows:

Value	Description
ringing	The user of the DECT phone gets an incoming call.
calling	The user of the DECT phone initiates an outgoing call.
paging	The DECT phone is going to be searched in the DECT area.
connected	The user of the DECT phone is connected with a partner.

idle	The user of the DECT phone is not connected to a partner.
none	A user is not assigned to this DECT phone.

3.10 PPUserSIPEventType

This type describes the SIP state of a DECT phone

PPUserSIPEventType is an enumeration value defined as follows:

Value	Description
sipRegistrationEvent	For the user of the DECT phone a SIP registration is started.
sipRegistrationEndEvent	For the user of the DECT phone a SIP registration is ended. The result is contained in <i>PPSIPStateType</i> .
sipNotifyEvent	For the user of the DECT phone a SIP notification event is received.

3.11 AccountStateType

This type contains all states of an account. It is used in *OpenResp* defined in this document.

Name	Type	Description
id	Int	Identifies the account: 0 → user account 1 → admin account 2 → root account
username	string	The user name of this account
state	string	One of <i>dontCare</i> , <i>changed</i> , <i>notChanged</i> , <i>expired</i> .

3.12 VersionType

This type contains the element version data field. It is used in *GetVersions*.

Name	Type	Description
Element	string	Version element string, one of ' <i>ActivateRemoteSystemDump</i> ' ' <i>AddUserToConference</i> ' ' <i>CancelMessage</i> ' ' <i>ChangeUserInConference</i> ' ' <i>CreateAccount</i> ' ' <i>CreateACLEntry</i> ' ' <i>CreateAlarmTrigger</i> ' ' <i>CreateBluetoothBeacon</i> ' ' <i>CreateConference</i> ' ' <i>CreateDigitTreatment</i> ' ' <i>CreateFixedPP</i> ' ' <i>CreateLDAP</i> '

		'CreatePPDev' 'CreatePPUser' 'CreateRFP' 'CreateSite' 'CreateWLANProfile' 'CreateXMLApplication' 'DBBackupToUSB' 'DBRestoreFromUSB' 'DeleteAccount' 'DeleteACLEntry' 'DeleteAlarmTrigger' 'DeleteBluetoothBeacon' 'DeleteConference' 'DeleteDigitTreatment' 'DeleteEventLogBuffer' 'DeleteLDAP' 'DeleteMessage' 'DeletePPDev' 'DeletePPUser' 'DeleteRFP' 'DeleteRFPCaptureList' 'DeleteRFPCaptureListElem' 'DeleteSite' 'DeleteUserFromConference' 'DeleteWLANProfile' 'DeleteXMLApplication' 'EventAccountCnf' 'EventAccountSummary' 'EventACLEnf' 'EventAddUserToConference' 'EventAdvancedSIPCnf' 'EventAlarmCallProgress' 'EventAlarmTriggerCnf' 'EventAlarmTrigger' 'EventAutoDBBackupCnf' 'EventAutoDBBackupFileNameCnf' 'EventBasicSIPCnf' 'EventBackupSIPCnf' 'EventBluetoothClientStatistic' 'EventBluetoothBeaconCnf' 'EventBluetoothBeaconSummary' 'EventBluetoothGlobalSettingsCnf' 'EventBluetoothSensitivityCnf' 'EventChangeUserInConference' 'EventCreateConference' 'EventConferenceRelease' 'EventConferenceRequest' 'EventConferenceRoomCnf' 'EventConferenceServerSIPCnf' 'EventConfigURLCnf' 'EventCoreDumpURLCnf'
--	--	---

		'EventDbTransferState' 'EventDECTAuthCodeCnf' 'EventDECTEncryptionCnf' 'EventDECTPagingAreaSizeCnf' 'EventDECTPowerLimitCnf' 'EventDECTPpSettingsCnf' 'EventDECTRegDomainCnf' 'EventDECTSubscriptionMode' 'EventDevAutoCreateCnf' "EventDeleteConference" "EventDeleteUserFromConference" 'EventDigitTreatmentCnf' 'EventDTMFCnf' 'EventEULAConfirmCnf' 'EventEventLogEntry' 'EventFACCnf' 'EventFACPrefixCnf' 'EventFreeConferenceChannels' 'EventHealthState' 'EventIMACnf' 'EventIntercomCallHandlingSIPCnf' 'EventLDAPCnf' 'EventLicenseCnf' 'EventLicensedCodecLines'EventLicenseFile' 'EventLicenseServerListCnf' 'EventMessageConfirmation' 'EventMessageProgress' 'EventMessageQueueEmpty' 'EventMessageSend' 'EventNetParamsCnf' 'EventNTPServerCnf' 'EventOMMCertificateCnf' 'EventOMPURLCnf' 'EventPARKCnf' 'EventPARKFromServerResult' 'EventPBXEnrolmentCnf' 'EventPermissionChange' 'EventPortRangeSIPCnf' 'EventPositionHistory' 'EventPositionInfo' 'EventPositionRequest' 'EventPositionTrack' 'EventPpProfileCnf' 'EventPPDevCnf' 'EventPPDevSummary' "EventPPFirmwareUpdateCnf" 'EventPPFirmwareUpdateOverview' 'EventPPFirmwareUpdateStatus' 'EventPPFirmwareURLCnf' 'EventPPLoginVariantCnf' 'EventPPState'
--	--	--

		<p> <i>'EventPPTTransaction'</i> <i>'EventPPUserCnf'</i> <i>'EventPPUserSummary'</i> <i>'EventPreserveUserDeviceRelationCnf'</i> <i>'EventReadyForConferencing'</i> <i>'EventSARICnf'</i> <i>'EventRegistrationTrafficShapingCnf'</i> <i>'EventRemoteAccessCnf'</i> <i>'EventRemoteSystemDumpCnf'</i> <i>'EventRestrictedSubscriptionDurationCnf'</i> <i>'EventRFPCapture'</i> <i>'EventRFPCaptureCnf'</i> <i>'EventRFPCnf'</i> <i>'EventRFPCConnectAttempt'</i> <i>'EventRFPDDetails'</i> <i>'EventRFPIpQuality'</i> <i>'EventRFPMCnf'</i> <i>'EventRFPMediaStreamQuality'</i> <i>'EventRFPState'</i> <i>'EventRFPSummary'</i> <i>'EventRFPsSyncQuality'</i> <i>'EventRFPsSyncRel'</i> <i>'EventRTPConferenceStreamChg'</i> <i>'EventRTPCnf'</i> <i>'EventSecureOMMCertificateServerImportCnf'</i> <i>'EventSecurePROVCertificateServerImportCnf'</i> <i>'EventSecureSIPCertificateCnf'</i> <i>'EventSecureSIPCertificateServerImportCnf'</i> <i>'EventSecureSIPCnf'</i> <i>'EventSiteCnf'</i> <i>'EventSiteSummary'</i> <i>'EventSoftwareImageURLCnf'</i> <i>'EventSpecialBrandingCnf'</i> <i>'EventSNMPCnf'</i> <i>'EventStbStateChange'</i> <i>'EventSuplServCnf'</i> <i>'EventSyslogServerCnf'</i> <i>'EventSystemCredentialsCnf'</i> <i>'EventSystemNameCnf'</i> <i>'EventSystemProvUpdTrigCnf'</i> <i>'EventSysToneSchemeCnf'</i> <i>'EventSysVoiceboxNumCnf'</i> <i>'EventTimeZoneCnf'</i> <i>'EventTimeZoneDetailsCnf'</i> <i>'EventTimeZoneList'</i> <i>'EventUsedConfigURL'</i> <i>'EventUserDataImport'</i> <i>'EventUserDataServerCnf'</i> <i>'EventUserDeviceSyncOMMCnf'</i> <i>'EventUserMonitoringCnf'</i> <i>'EventWLANClient'</i> </p>
--	--	--

		'EventWLANProfileCnf' 'EventWLANRegDomainCnf' 'EventXMLApplicationCnf' 'GenerateHealthStateAlarmTriggers' 'GetAccount' 'GetAccountSummary' 'GetACLEntry' 'GetActivePPDev' 'GetAdvancedSIP' 'GetAlarmTrigger' 'GetAlarmTriggerSummary' 'GetAutoDBBackup' 'GetAutoDBBackupFileName' 'GetBasicSIP' 'GetBackupSIP' 'GetBluetoothClientStatistic' 'GetBluetoothBeacon' 'GetBluetoothBeaconSummary' 'GetBluetoothGlobalSettings' 'GetBluetoothSensitivity' 'GetConferenceRoom' 'GetConferenceServerSIP' 'GetConfigURL' 'GetCoreDumpURL' 'GetDbTransferState' 'GetDECTAuthCode' 'GetDECTEncryption' 'GetDECTPagingAreaSize' 'GetDECTPowerLimit' 'GetDECTPpSettings' 'GetDECTRegDomain' 'GetDECTSubscriptionMode' 'GetDevAutoCreate' 'GetDigitTreatment' 'GetDigitTreatmentSummary' 'GetDTMF' 'GetEULAConfirm' 'GetEventLogBuffer' 'GetFACLlist' 'GetFACPrefix' 'GetFile' 'GetFlashMemUsage' 'GetFreeConferenceChannels' 'GetG729ChannelsForConference' 'GetHealthState' 'GetIMA' 'GetIntercomCallHandlingSIP' 'GetLastPPDevAction' 'GetLDAP' 'GetLicense' 'GetLicenseServerList'
--	--	---

		'GetLicensedCodeclines' 'GetNetParams' 'GetNTPServer' 'GetOMPURL' 'GetOMMCertificate' "GetPARK" 'GetPBXEnrolment' 'GetPortRangeSIP' GetPpProfile' 'GetPPDev' 'GetPPDevSummary' 'GetPPFirmwareUpdate' 'GetPPFirmwareUpdateOverview' 'GetPPFirmwareUpdateStatus' 'GetPPFirmwareURL' 'GetPPLoginVariant' 'GetPPState' 'GetPPUser' 'GetPPUserSummary' 'GetPreserveUserDeviceRelation' 'GetPublicKey' 'GetReadyForConferencing' 'GetRegistrationTrafficShaping' 'GetRemoteAccess' 'GetRemoteSystemDump' 'GetRestrictedSubscriptionDuration' 'GetRFP' 'GetRFPCapture' 'GetRFPCaptureList' 'GetRFPIpQuality' 'GetRFPM' "GetRFPMediaStreamQuality" 'GetRFPStatistic' 'GetRFPStatisticConfig' 'GetRFPSummary' 'GetRFPsync' 'GetRFPsyncQuality' 'GetRTP' 'GetSARI' 'GetSecureSIP' 'GetSecureOMMCertificate' 'GetSecureOMMCertificateServerImport' 'GetSecurePROVCertificateServerImport' 'GetSecureSIPCertificate' 'GetSecureSIPCertificateServerImport' 'GetSite' 'GetSiteSummary' 'GetSNMP' 'GetSoftwareImageURL' 'GetSpecialBranding' 'GetStbState'
--	--	--

		<ul style="list-style-type: none"> 'GetSuplServ' 'GetSyslogServer' 'GetSysStatisticConfig' 'GetSysStatisticMinMax' 'GetSysStatisticMinMaxRecord' 'GetSysStatisticMinMaxSummary' 'GetSysStatisticOccurrence' 'GetSystemCredentials' 'GetSystemName' 'GetSystemProvUpdTrig' 'GetSysToneScheme' 'GetSysVoiceboxNum' 'GetTimeZone' 'GetTimeZoneDetails' 'GetTimeZoneList' 'GetUsedConfigURL' 'GetUserDataServer' 'GetUserDeviceSyncOMM' 'GetUserMonitoring' 'GetVersions' 'GetWLANClients' 'GetWLANProfile' 'GetWLANRegDomain' 'GetWLANRegDomainList' 'GetXMLApplication' 'Limits' 'ManualDBBackup' 'ManualDBRestore' 'ManualUserDataImport' 'Open' 'PARKFromServer' 'Ping' 'RequestPositionInfo' 'PutFile' 'ResetRFPMediaStreamQuality' 'ResetRFPStatistic' 'ResetSysStatistic' 'ResetTimeZoneDetails' 'SendMessage' 'SetAccount' 'SetACLEntry' 'SetAdvancedSIP' 'SetAlarmTrigger' 'SetAutoDBBackup' 'SetBasicSIP' 'SetBackupSIP' 'SetBluetoothBeacon' 'SetBluetoothGlobalSettings' 'SetBluetoothSensitivity' 'SetConferenceRoom' 'SetConferenceServerSIP'
--	--	--

		' <i>SetConfigURL</i> ' ' <i>SetCoreDumpURL</i> ' ' <i>SetDECTAuthCode</i> ' ' <i>SetDECTEncryption</i> ' ' <i>SetDECTPagingAreaSize</i> ' ' <i>SetDECTPowerLimit</i> ' ' <i>SetDECTPpSettings</i> ' ' <i>SetDECTRegDomain</i> ' ' <i>SetDECTSubscriptionMode</i> ' ' <i>SetDevAutoCreate</i> ' ' <i>SetDigitTreatment</i> ' ' <i>SetDTMF</i> ' ' <i>SetEULAConfirm</i> ' ' <i>SetFAC</i> ' ' <i>SetFACList</i> ' ' <i>SetFACPrefix</i> ' ' <i>SetIMA</i> ' ' <i>SetIntercomCallHandlingSIP</i> ' ' <i>SetLDAP</i> ' ' <i>SetLicenseServerList</i> ' ' <i>SetNetParams</i> ' ' <i>SetOMMCertificate</i> ' ' <i>SetOMPURL</i> ' ' <i>SetNTPServer</i> ' ' <i>SetPARK</i> ' ' <i>SetPBXEnrolment</i> ' ' <i>SetPortRangeSIP</i> ' ' <i>SetPpProfile</i> ' ' <i>SetPPDev</i> ' ' <i>SetPPFirmwareUpdate</i> ' ' <i>SetPPFirmwareURL</i> ' ' <i>SetPPLginVariant</i> ' ' <i>SetPPUser</i> ' ' <i>SetPPUserDevRelation</i> ' ' <i>SetPPUserTracking</i> ' ' <i>SetPreserveUserDeviceRelation</i> ' ' <i>SetRegistrationTrafficShaping</i> ' ' <i>SetRemoteAccess</i> ' ' <i>SetRemoteSystemDump</i> ' ' <i>SetRestrictedSubscriptionDuration</i> ' ' <i>SetRFP</i> ' ' <i>SetRFPCapture</i> ' ' <i>SetRFPM</i> ' ' <i>SetRTP</i> ' ' <i>SetRTPConferenceStreamChg</i> ' ' <i>SetSARI</i> ' ' <i>SetSecureSIP</i> ' ' <i>SetSecureOMMCertificate</i> ' ' <i>SetSecureOMMCertificateServerImport</i> ' ' <i>SetSecurePROVCertificateServerImport</i> ' ' <i>SetSecureSIPCertificate</i> '
--	--	--

		<i>'SetSecureSIPCertificateServerImport'</i> <i>'SetSite'</i> <i>'SetSNMP'</i> <i>'SetSoftwareImageURL'</i> <i>'SetSpecialBranding'</i> <i>'SetSuplServ'</i> <i>'SetSyslogServer'</i> <i>'SetSystemCredentials'</i> <i>'SetSystemName'</i> <i>'SetSystemProvUpdTrig'</i> <i>'SetSysToneScheme'</i> <i>'SetSysVoiceboxNum'</i> <i>'SetTimeZone'</i> <i>'SetTimeZoneDetails'</i> <i>'SetUserDataServer'</i> <i>'SetUserDeviceSyncOMM'</i> <i>"SetUserMonitoring"</i> <i>'SetWLANProfile'</i> <i>'SetWLANRegDomain'</i> <i>'SetXMLApplication'</i> <i>'SoftwareUpate'</i> <i>'Subscribe'</i> <i>'SystemRestart'</i>
--	--	--

3.13 PublicKeyType

This type contains the element data fields of the public key. It is used in OpenResp.

Name	Type	Description
modulus	string	Hexadecimal encoded public key modulus n
exponent	string	Hexadecimal encoded public key exponent e

4 Messages

All messages used in the OM AXI are described in this chapter.

4.1 Basic Requests

4.1.1 Open

This request is sent from the client to the OMM.

This is the only request the OMM accepts on a freshly set up TCP link, which has the state *new*. If *Open* fails, the client may send additional *Open* messages with different ingredients (e. g. a different OM AXI protocol version or password) to try to set up a session. The link remains in the state *new* as long as *Open* was not successful.

Open is not accepted by the OMM on an already opened session.

Following fields are defined for the request *Open* in addition to the common attributes:

Name	Type	Mandatory	Description
username	string	yes	User name for authentication
password	string	yes	Password for authentication

The response to this request does also contain information about the OMM standby feature. If the client received the error *EFailed*, it should connect to the other, active OMM which's IP address is contained in the attribute *ommStbAddr*.

The reply is an object called *OpenResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
protocolVersion	integer	yes	This is the OM AXI protocol version. This OM AXI specification supports <i>protocolVersion="44"</i> <i>The protocolVersion changes when at least one minor or major version number changes (see 4.1.2).</i>
ommVersion	string	no	Full version string of the OMM: <majorRelease>.<minorRelease>.{RC x SP y <bugfixVersion>} [Build z] [version description], e. g. „2.1 RC4“, „2.1 SP1“, „2.1.5“, „2.1SP1 Build 2“,r „3.0 RC1 Build 1 - OpenMobility SIP 3.0RC1-Internal“. RCx: Release candidate number x SPy: Release service pack number y bugfixVersion: Old bugfix release version number Build z: Build release number z for internal use
ommAxiSpecVersion	string	no	Version string of the used OM AXI specification (e. g. “6.1”).
axiClients	integer	no	Number of currently connected AXI clients

minPPSwVersion1	string	yes	Minimal Mitel DECT Phone SW version for Mitel DECT Phone 600 family required to run with this OMM version.
minPPSwVersion2	string	yes	Minimal Mitel DECT Phone SW version for Mitel DECT Phone 6xx family required to run with this OMM version.
accountState	sequence of accountStateType	no	States of all (user, admin, root) account data currently configured.
permission	sequence of PermissionType	no	All Permissions given to this client. They may depend on the user account used for authentication and on the current license state. Only returned if <i>Open</i> succeeded.
ommStbState	enumeration	yes	State of the OMM standby feature. One of: "OK": OMM standby running and active "NotSynchronized": OMM standby active but the standby OMM is not visible or the database is not synchronized "DifferentOMMTypes", "DifferentOMMVersions": OMM standby active, but OMMs do not fit together "None" no OMM standby active
ommStbAddr	string	no	In case of OMM standby is active: IP address of the other OMM
ommStream	string	no	The used OMM stream – e. g. "ffsip", "a5000", "ocx", "oc10xx", ... Only for information! No OMM properties should be conveyed from this string!
ommPlatform	string	no	The used platform the OMM is running on – e. g. "linux-pc", "rfp", ... Only for information! No OMM properties should be conveyed from this string!
uptime	integer	no	Up time since last OMM start in seconds.
EULAConfirm	boolean	no	„1” or “true”, if the EULA has been confirmed already.
publicKey	PublicKeyType	yes	Public key
haveAdditionalSettings	boolean, default=false	no	“1” or “true”, if additional settings are supported (see 4.64)
haveAutoDB	boolean, default=false	no	“1” or “true”, if automatic database backup and restore is supported
haveBluetooth	boolean, default=false	no	“1” or “true”, if Bluetooth dongle configuration is

			supported
haveConfigPython	boolean, default=false	no	"1" or "true", if internal python configuration is supported.
haveConference	boolean, default=false	no	"1" or "true", if a conference server is supported
haveDECTPpSettings	boolean, default=false	no	"1" or "true", if DECT DECT phone configuration is supported
haveDECTRegDomain	boolean, default=false	no	"1" or "true", if DECT regulatory domain configuration is supported
haveDigitTreatment	boolean, default=false	no	"1" or "true", if extended digit treatment configuration is supported
haveDigitTreatmentSimple	boolean, default=false	no	"1" or "true", if simple digit treatment configuration is supported
haveLDAP	boolean, default=false	no	"1" or "true", if an extended corporate LDAP directory access is supported
havePPFirmwareUpdate	boolean, default=false	no	"1" or "true", if DECT phone firmware update is supported
haveEnrolmentPBX	boolean, default=false	no	"1" or "true", if PBX enrolment configuration is supported
haveEnrolmentPP	boolean, default=false	no	"1" or "true", if DECT phone enrolment configuration is supported
haveEnrolmentRFP	boolean, default=false	no	"1" or "true", if RFP enrolment configuration is supported
haveExternalUserData	boolean, default=false	no	"1" or "true", if external user data provisioning configuration is supported
haveFACs	boolean, default=false	no	"1" or "true", if feature access code configuration is supported
haveHCM	boolean, default=false	no	"1" or "true", if DECT phone profile configuration is supported.
haveIMA	boolean, default=false	no	"1" or "true", if integrated message and alarm server application (IMA) configuration is supported
haveIPParamCnf	boolean, default=false	no	"1" or "true", if special IP parameter configuration (TOS, VLAN) is supported
haveLicensing	boolean, default=false	no	"1" or "true", if license configuration is supported
haveLicensingCodec	boolean, default=false	no	"1" or "true", if licensed G.729 codec handling is supported

haveLocating	boolean, default=false	no	“1” or “true”, if LOCATING is supported: Note: Different to OMM release 2.0 this does not mean that the external licensed location application is admitted to run! The external licensed location application has to analyze the event “EventLicenseCnf” to decide.
haveNTP	boolean, default=false	no	“1” or “true”, if NTP server configuration is supported.
haveOmmLogForward	boolean, default=false	no	“1” or “true”, if forwarding of loggings from OMM running on a Linux PC to syslog server is supported.
haveOMP	boolean, default=false	no	“1” or “true”, if OMP Java Web start configuration is supported
havePagingAreas	boolean, default=false	no	“1” or “true”, if paging areas configuration is supported
havePARKFile	boolean, default=false	no	“1” or “true”, if PARK support from an external server is supported (incl. upload and download of related files).
haveProtFTP	boolean, default=false	no	“1” or “true”, if FTP protocol type is supported in <i>ProtocolType</i> .
haveProtFTPS	boolean, default=false	no	“1” or “true”, if FTPS protocol type is supported in <i>ProtocolType</i> .
haveProtHTTP	boolean, default=false	no	“1” or “true”, if HTTP protocol type is supported in <i>ProtocolType</i> .
haveProtHTTPS	boolean, default=false	no	“1” or “true”, if HTTPS protocol type is supported in <i>ProtocolType</i> .
haveProtSCP	boolean, default=false	no	“1” or “true”, if SCP protocol type is supported in <i>ProtocolType</i> .
haveProtSFTP	boolean, default=false	no	“1” or “true”, if SFTP protocol type is supported in <i>ProtocolType</i> .
haveProtTFTP	boolean, default=false	no	“1” or “true”, if TFTP protocol type is supported in <i>ProtocolType</i> .
haveRFPMcnf	boolean, default=false	no	“1” or “true”, if RFP management configuration is supported
haveRFPOMM	boolean, default=false	no	“1” or “true”, if the OMM is running on an RFP.
haveRTPPortBaseCnf	boolean, default=false	no	“1” or “true”, if RTP port base configuration is supported
haveSARI	boolean, default=false	no	“1” or “true”, if DECT SARI handling is

			supported
haveSemiAttendedTransfer	boolean, default=false	no	"1" or "true", if SIP semi attended transfer configuration is supported.
haveSIP	boolean, default=false	no	"1" or "true", if SIP configuration is supported.
haveSNMP	boolean, default=false	no	"1" or "true", if SNMP configuration is supported
haveSoftwareImageURL	boolean, default=false	no	"1" or "true", if software image url configuration is supported
haveSOSNumberPP	boolean, default=false	no	"1" or "true", if SOS number for a DECT phone configuration is supported
haveSpecialBranding	boolean, default=false	no	"1" or "true", if special branding for the Mitel SIP-DECT product is supported
haveSRTP	boolean, default=false	no	"1" or "true", if SRTP configuration is supported.
haveTZones	boolean, default=false	no	"1" or "true", if time zone configuration is supported
haveUnboundDevices	boolean, default=false	no	"1" or "true", if unbound device configuration is supported
haveUMO	boolean, default=false	no	"1" or "true", if user monitoring is supported
haveUserDeviceSync	boolean, default=false	no	"1" or "true", if user device synchronization between OMM sites for roaming is supported
haveVLAN	boolean, default=false	no	"1" or "true", if VLAN priority configuration is supported
haveVideo	boolean, default=false	no	"1" or "true", if video is supported
haveVoiceboxNumber	boolean, default=false	no	"1" or "true", if voice box number configuration is supported
haveWLAN	boolean, default=false	no	"1" or "true", if WLAN configuration is supported
haveXML	boolean, default=false	no	"1" or "true", if built-in XML application configuration is supported
haveXMLDynamic	boolean, default=false	no	"1" or "true", if additional/non built-in XML application configuration is supported
haveXMLCorpDir	boolean, default=false	no	"1" or "true", if XML corporate directory application is supported

Possible values for *errCode*:

Value	Description
EForbidden	This OMM is not configured in standby mode but not ready to be connected.

EFailed	This OMM is inactive and another standby OMM is active
EAuth	An authentication error occurred for this AXI cclient connection attempt
ENoMem	The maximal number of AXI client connections is already reached
EExist	An AXI client connection for this user is already established (some AXI clients can only connect once)
ENoEnt	The AXI interface is disabled for external clients

For further *errCode* values see 3.3.

4.1.2 GetVersions

This request is sent from the client to the OMM to request the versions of requests/responses or events.

All version strings have the following structure and meaning:

```
e. g.
<PutFile
  = „1.0.0”/> /* <Major>.<Minor>.<Optional/Bugfix>*/
  |
  | |
  | | - „Optional/Bugfixes”
  | |   +=1 for expansion of optional attributes
  | |   +=1 for functional changes based on bug fixes
  | |
  | | --- „Minor”
  | |     +=1 for changes of attribute ranges
  | |     +=1 for new Enum values of attributes [or new
  | |       choice values (flags)]
  | |     +=1 for new error codes
  | |     Changing minor number forces “0” for optional/bugfix number
  | |
  | | ----- „Major”
  | |           +=1 for expansion of mandatory attributes
  | |           +=1 for element delete or changes
  | |           +=1 for attribute delete or changes
  | |           +=1 for optional/mandatory changes of an attribute
  | |           Changing major number forces “0” for minor and
  | |             optional/bugfix number
```

When at least one minor or major number of a version changes, the OMM AXI *protocolVersion* changes (see 4.1.1):

If an element is not supported (wrong or deleted) by OM AXI the version string “0.0.0” is used for this element in the response.

Following fields are defined for the request *GetVersions* in addition to the common attributes:

Name	Type	Mandatory	Description
version	sequence of VersionType	no	All elements the client wants to get version information for. If no element was given, all versions will be delivered.

The reply is an object called *GetVersionsResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
ActivateRemoteSystemDump	string	no	Version string of the request/response elements 'ActivateRemoteSystemDump'.

AddUserToConference	string	no	Version string of the request/response elements 'AddUserToConference'.
CancelMessage	string	no	Version string of the request/response elements 'CancelMessage'.
ChangeUserInConference	string	no	Version string of the request/response elements 'ChangeUserInConference'.
CreateAccount	string	no	Version string of the request/response elements 'CreateAccount'.
CreateACLEntry	string	no	Version string of the request/response elements 'CreateACLEntry'.
CreateAlarmTrigger	string	no	Version string of the request/response elements 'CreateAlarmTrigger'.
CreateBluetoothBeacon	string	no	Version string of the request/response elements 'CreateBluetoothBeacon'.
CreateConference	string	no	Version string of the request/response elements 'CreateConference'.
CreateDigitTreatment	string	no	Version string of the request/response elements 'CreateDigitTreatment'.
CreateFixedPP	string	no	Version string of the request/response elements 'CreateFixedPP'.
CreateLDAP	string	no	Version string of the request/response elements 'CreateLDAP'.
CreatePPDev	string	no	Version string of the request/response elements 'CreatePPDev'.
CreatePPUser	string	no	Version string of the request/response elements 'CreatePPUser'.
CreateRFP	string	no	Version string of the request/response elements 'CreateRFP'.
CreateSite	string	no	Version string of the request/response elements 'CreateSite'.
CreateVideoDev	string	no	Version string of the request/response elements 'CreateVideoDev'.
CreateWLANProfile	string	no	Version string of the request/response elements 'CreateWLANProfile'.
CreateXMLApplication	string	no	Version string of the request/response elements 'CreateXMLApplication'.
DBBackupToUSB	string	no	Version string of the request/response

			elements 'DBBackupToUSB'.
DBRestoreFromUSB	string	no	Version string of the request/response elements 'DBRestoreFromUSB'.
DeleteAccount	string	no	Version string of the request/response elements 'DeleteAccount'.
DeleteACLEntry	string	no	Version string of the request/response elements 'DeleteACLEntry'.
DeleteAlarmTrigger	string	no	Version string of the request/response elements 'DeleteAlarmTrigger'.
DeleteBluetoothBeacon	string	no	Version string of the request/response elements 'DeleteBluetoothBeacon'.
DeleteConference	string	no	Version string of the request/response elements 'DeleteConference'.
DeleteDigitTreatment	string	no	Version string of the request/response elements 'DeleteDigitTreatment'.
DeleteEventLogBuffer	string	no	Version string of the request/response element 'DeleteEventLogBuffer'.
DeleteLDAP	string	no	Version string of the request/response elements 'DeleteLDAP'.
DeleteMessage	string	no	Version string of the request/response elements 'DeleteMessage'.
DeletePPDev	string	no	Version string of the request/response elements 'DeletePPDev'.
DeletePPUser	string	no	Version string of the request/response elements 'DeletePPUser'.
DeleteRFP	string	no	Version string of the request/response elements 'DeleteRFP'.
DeleteRFPCaptureList	string	no	Version string of the request/response elements 'DeleteRFPCaptureList'.
DeleteRFPCaptureListElem	string	no	Version string of the request/response elements 'DeleteRFPCaptureListElem'.
DeleteSite	string	no	Version string of the request/response elements 'DeleteSite'.
DeleteUserFromConference	string	no	Version string of the request/response elements 'DeleteUserFromConference'.
DeleteVideoDev	string	no	Version string of the request/response elements 'DeleteVideoDev'.

DeleteWLANProfile	string	no	Version string of the request/response elements 'DeleteWLANProfile'.
DeleteXMLApplication	string	no	Version string of the request/response elements 'DeleteXMLApplication'.
EventAccountCnf	string	no	Version string of the event element 'EventAccountCnf'.
EventAccountSummary	string	no	Version string of the event element 'EventAccountSummary'.
EventACLCnf	string	no	Version string of the event element 'EventACLCnf'.
EventAddUserToConference	string	no	Version string of the element 'EventAddUserToConference'.
EventAdvancedSIPCnf	string	no	Version string of the event element 'EventAdvancedSIPCnf'.
EventAlarmCallProgress	string	no	Version string of the event element 'EventAlarmCallProgress'.
EventAlarmTrigger	string	no	Version string of the event element 'EventAlarmTrigger'.
EventAlarmTriggerCnf	string	no	Version string of the event element 'EventAlarmTriggerCnf'.
EventAutoDBBackupCnf	string	no	Version string of the event element 'EventAutoDBBackupCnf'.
EventAutoDBBackupFileNameCnf	string	no	Version string of the event element 'EventAutoDBBackupFileNameCnf'.
EventBasicSIPCnf	string	no	Version string of the event element 'EventBasicSIPCnf'.
EventBackupSIPCnf	string	no	Version string of the event element 'EventBackupSIPCnf'.
EventBluetoothClientStatistic	string	no	Version string of the element 'EventBluetoothClientStatistic'.
EventBluetoothBeaconCnf	string	no	Version string of the event element 'EventBluetoothBeaconCnf'.
EventBluetoothBeaconSummary	string	no	Version string of the request/response elements 'EventBluetoothBeaconSummary'.
EventBluetoothGlobalSettingsCnf	string	no	Version string of the event element 'EventBluetoothGlobalSettingsCnf'.
EventBluetoothSensitivityCnf	string	no	Version string of the event element

			'EventBluetoothSensitivityCnf'.
EventChangeUserInConference	string	no	Version string of the element 'EventChangeUserInConference'.
EventCreateConference	string	no	Version string of the element 'EventCreateConference'.
EventConferenceRelease	string	no	Version string of the element 'EventConferenceRelease'.
EventConferenceRequest	string	no	Version string of the element 'EventConferenceRequest'.
EventConferenceRoomCnf	string	no	Version string of the element 'EventConferenceRoomCnf'.
EventConferenceServerSIP	string	no	Version string of the element 'EventConferenceServerSIP'.
EventConfigURLCnf	string	no	Version string of the event element 'EventConfigURLCnf'.
EventCoreDumpURLCnf	string	no	Version string of the event element 'EventCoreDumpURLCnf'.
EventDbTransferState	string	no	Version string of the event element 'EventDbTransferState'.
EventDECTAuthCodeCnf	string	no	Version string of the event element 'EventDECTAuthCodeCnf'.
EventDECTEncryptionCnf	string	no	Version string of the event element 'EventDECTEncryptionCnf'.
EventDECTPagingAreaSizeCnf	string	no	Version string of the event element 'EventDECTPagingAreaSizeCnf'.
EventDECTPowerLimitCnf	string	no	Version string of the event element 'EventDECTPowerLimitCnf'.
EventDECTPpSettingsCnf	string	no	Version string of the event element 'EventDECTPpSettingsCnf'.
EventDECTRegDomainCnf	string	no	Version string of the event element 'EventDECTRegDomainCnf'.
EventDECTSubscriptionMode	string	no	Version string of the event element 'EventDECTSubscriptionMode'.
EventDeleteConference	string	no	Version string of the element 'EventDeleteConference'.
EventDeleteUserFromConference	string	no	Version string of the element 'EventDeleteUserFromConference'.

EventDevAutoCreateCnf	string	no	Version string of the event element 'EventDevAutoCreateCnf'.
EventDigitTreatmentCnf	string	no	Version string of the event element 'EventDigitTreatmentCnf'.
EventDTMFCnf	string	no	Version string of the event element 'EventDTMFCnf'.
EventEULAConfirmCnf	string	no	Version string of the event element 'EventEULAConfirmCnf'.
EventEventLogEntry	string	no	Version string of the event element 'EventEventLogEntry'.
EventFACCnf	string	no	Version string of the event element 'EventFACCnf'.
EventFACPrefixCnf	string	no	Version string of the event element 'EventFACPrefixCnf'.
EventFreeConferenceChannels	string	no	Version string of the element 'EventFreeConferenceChannels '.
EventHealthState	string	no	Version string of the event element 'EventHealthState'.
EventIMACnf	string	no	Version string of the event element 'EventIMACnf'.
EventIntercomCallHandlingSIPCnf	string	no	Version string of the event element 'EventIntercomCallHandlingSIPCnf'.
EventLDAPCnf	string	no	Version string of the event element 'EventLDAPCnf'.
EventLicenseCnf	string	no	Version string of the event element 'EventLicenseCnf'.
EventLicensedCodecLines	string	no	Version string of the event element 'EventLicensedCodecLines'.
EventLicenseFile	string	no	Version string of the event element 'EventLicenseFile'.
EventLicenseServerListCnf	string	no	Version string of the event element 'EventLicenseServerListCnf'.
EventMessageConfirmation	string	no	Version string of the event element 'EventMessageConfirmation'.
EventMessageProgress	string	no	Version string of the event element 'EventMessageProgress'.
EventMessageQueueEmpty	string	no	Version string of the event element

			'EventMessageQueueEmpty'.
EventMessageSend	string	no	Version string of the event element 'EventMessageSend'.
EventNTPServerCnf	string	no	Version string of the event element 'EventNTPServerCnf'.
EventOMMCertificateCnf	string	no	Version string of the request/response elements 'EventOMMCertificateCnf'.
EventOMPURLCnf	string	no	Version string of the event element 'EventOMPURLCnf'.
EventNetParamsCnf	string	no	Version string of the event element 'EventNetParamsCnf'.
EventPARKCnf	string	no	Version string of the event element 'EventPARKCnf'.
EventPARKFromServerResult	string	no	Version string of the request/response elements 'EventPARKFromServerResult'.
EventPBXEnrolmentCnf	string	no	Version string of the event element 'EventPBXEnrolmentCnf'.
EventPermissionChange	string	no	Version string of the event element 'EventPermissionChange'.
EventPortRangeSIPCnf	string	no	Version string of the event element 'EventPortRangeSIPCnf'.
EventPositionHistory	string	no	Version string of the event element 'EventPositionHistory'.
EventPositionInfo	string	no	Version string of the event element 'EventPositionInfo'.
EventPositionRequest	string	no	Version string of the event element 'EventPositionRequest'.
EventPositionTrack	string	no	Version string of the event element 'EventPositionTrack'.
EventPpProfileCnf	string	no	Version string of the event element 'EventPpProfileCnf'.
EventPPDevCnf	string	no	Version string of the event element 'EventPPDevCnf'.
EventPPDevSummary	string	no	Version string of the event element 'EventPPDevSummary'.
EventPPFirmwareUpdateCnf	string	no	Version string of the event element 'EventPPFirmwareUpdateCnf'.

EventPPFirmwareUpdateOverview	string	no	Version string of the event element 'EventPPFirmwareUpdateOverview'.
EventPPFirmwareUpdateStatus	string	no	Version string of the event element 'EventPPFirmwareUpdateStatus'.
EventPPFirmwareURLCnf	string	no	Version string of the event element 'EventPPFirmwareURLCnf'.
EventPPLoginVariantCnf	string	no	Version string of the event element 'EventPPLoginVariantCnf'.
EventPPState	string	no	Version string of the event element 'EventPPState'.
EventPPTransaction	string	no	Version string of the event element 'EventPPTransaction'.
EventPPUserCnf	string	no	Version string of the event element 'EventPPUserCnf'.
EventPPUserSummary	string	no	Version string of the event element 'EventPPUserSummary'.
EventPreserveUserDeviceRelationCnf	string	no	Version string of the event element 'EventPreserveUserDeviceRelationCnf'.
EventReadyForConferencing	string	no	Version string of the element 'EventReadyForConferencing'.
EventRegistrationTrafficShapingCnf	string	no	Version string of the event element 'EventRegistrationTrafficShapingCnf'.
EventRemoteAccessCnf	string	no	Version string of the event element 'EventRemoteAccessCnf'.
EventRemoteSystemDumpCnf	string	no	Version string of the event element 'EventRemoteSystemDumpCnf'.
EventRestrictedSubscriptionDurationCnf	string	no	Version string of the event element 'EventRestrictedSubscriptionDurationCnf'.
EventRFPCnf	string	no	Version string of the event element 'EventRFPCnf'.
EventRFPCapture	string	no	Version string of the event element 'EventRFPCapture'.
EventRFPCaptureCnf	string	no	Version string of the event element 'EventRFPCaptureCnf'.
EventRFPCConnectAttempt	string	no	Version string of the event element 'EventRFPCConnectAttempt'.
EventRFPDetails	string	no	Version string of the event element

			'EventRFPDetails'.
EventRFPipQuality	string	no	Version string of the event element 'EventRFPipQuality'.
EventRFPMCnf	string	no	Version string of the event element 'EventRFPMCnf'.
EventRFPMediaStreamQuality	string	no	Version string of the event element 'EventRFPMediaStreamQuality'.
EventRFPState	string	no	Version string of the event element 'EventRFPState'.
EventRFPSummary	string	no	Version string of the event element 'EventRFPSummary'.
EventRFPSyncQuality	string	no	Version string of the event element 'EventRFPSyncQuality'.
EventRFPSyncRel	string	no	Version string of the event element 'EventRFPSyncRel'.
EventRTPCnf	string	no	Version string of the event element 'EventRTPCnf'.
EventRTPConferenceStreamChg	string	no	Version string of the event element 'EventRTPConferenceStreamChg'.
EventSARICnf	string	no	Version string of the event element 'EventSARICnf'.
EventSecureOMMCertificateServerImportCnf	string	no	Version string of the request/response elements 'EventSecureOMMCertificateServerImportCnf'.
EventSecurePROVCertificateServerImportCnf	string	no	Version string of the request/response elements 'EventSecurePROVCertificateServerImportCnf'.
EventSecureSIPCertificateCnf	string	no	Version string of the request/response elements 'EventSecureSIPCertificateCnf'.
EventSecureSIPCertificateServerImportCnf	string	no	Version string of the request/response elements 'EventSecureSIPCertificateServerImportCnf'.
EventSecureSIPCnf	string	no	Version string of the event element 'EventSecureSIPCnf'.
EventSiteCnf	string	no	Version string of the event element 'EventSiteCnf'.
EventSiteSummary	string	no	Version string of the event element 'EventSiteSummary'.

EventSNMPCnf	string	no	Version string of the event element 'EventSNMPCnf'.
EventSoftwareImageURLCnf	string	no	Version string of the event element 'EventSoftwareImageURLCnf'.
EventSpecialBrandingCnf	string	no	Version string of the event element 'EventSpecialBrandingCnf'.
EventStbStateChange	string	no	Version string of the event element 'EventStbStateChange'.
EventSuplServCnf	string	no	Version string of the event element 'EventSuplServCnf'.
EventSyslogServerCnf	string	no	Version string of the event element 'EventSyslogServerCnf'.
EventSystemCredentialsCnf	string	no	Version string of the event element 'EventSystemCredentialsCnf'.
EventSystemNameCnf	string	no	Version string of the event element 'EventSystemNameCnf'.
EventSystemProvUpdTrigCnf	string	no	Version string of the event element 'EventSystemProvUpdTrigCnf'.
EventSysToneSchemeCnf	string	no	Version string of the event element 'eventSysToneSchemeCnf'.
EventSysVoiceboxNumCnf	string	no	Version string of the element event 'eventSysVoiceboxNumCnf'.
EventTimeZoneCnf	string	no	Version string of the event element 'EventTimeZoneCnf'.
EventTimeZoneDetailsCnf	string	no	Version string of the event element 'EventTimeZoneDetailsCnf'.
EventTimeZoneList	string	no	Version string of the event element 'EventTimeZoneList'.
EventUsedConfigURL	string	no	Version string of the event element 'EventUsedConfigURL'.
EventUserDataImport	string	no	Version string of the event element 'EventUserDataImport'.
EventUserDataServerCnf	string	no	Version string of the event element 'EventUserDataServerCnf'.
EventUserDeviceSyncOMMCnf	string	no	Version string of the event element 'EventUserDeviceSyncOMMCnf'.
EventUserMonitoringCnf	string	no	Version string of the event element

			'EventUserMonitoringCnf'.
EventVideoDevCnf	string	no	Version string of the event element 'EventVideoDevCnf'.
EventVideoDevLink	string	no	Version string of the event element 'EventVideoDevLink'.
EventVideoDevSummary	string	no	Version string of the event element 'EventVideoDevSummary'.
EventWLANClient	string	no	Version string of the event element 'EventWLANClient'.
EventWLANProfileCnf	string	no	Version string of the event element 'EventWLANProfileCnf'.
EventWLANRegDomainCnf	string	no	Version string of the event element 'EventWLANRegDomainCnf'.
EventXMLApplicationCnf	string	no	Version string of the event element 'EventXMLApplicationCnf'.
GenerateHealthStateAlarmTriggers	string	no	Version string of the request/response elements 'GenerateHealthStateAlarmTriggers'.
GetAccount	string	no	Version string of the request/response elements 'GetAccount'.
GetAccountSummary	string	no	Version string of the request/response elements 'GetAccountSummary'.
GetACLEntry	string	no	Version string of the request/response elements 'GetACLEntry'.
GetActivePPDev	string	no	Version string of the request/response elements 'GetActivePPDev'.
GetAdvancedSIP	string	no	Version string of the request/response elements 'GetAdvancedSIP'.
GetAlarmTrigger	string	no	Version string of the request/response elements 'GetAlarmTrigger'.
GetAlarmTriggerSummary	string	no	Version string of the request/response elements 'GetAlarmTriggerSummary'.
GetAutoDBBackup	string	no	Version string of the request/response elements 'GetAutoDBBackup'.
GetAutoDBBackupFileName	string	no	Version string of the request/response elements 'GetAutoDBBackupFileName'.
GetBasicSIP	string	no	Version string of the request/response elements 'GetBasicSIP'.

GetBackupSIP	string	no	Version string of the request/response elements 'GetBackupSIP'.
GetBluetoothClientStatistic	string	no	Version string of the request/response elements 'GetBluetoothClientStatistic'.
GetBluetoothBeacon	string	no	Version string of the request/response elements 'GetBluetoothBeacon'.
GetBluetoothBeaconSummary	string	no	Version string of the request/response elements 'GetBluetoothBeaconSummary'.
GetBluetoothGlobalSettings	string	no	Version string of the request/response elements 'GetBluetoothGlobalSettings'.
GetBluetoothSensitivity	string	no	Version string of the request/response elements 'GetBluetoothSensitivity'.
GetConfigURL	string	no	Version string of the request/response elements 'GetConfigURL'.
GetConferenceRoom	string	no	Version string of the request/response elements 'GetConferenceRoom'.
GetConferenceServerSIP	string	no	Version string of the request/response elements 'GetConferenceServerSIP'.
GetCoreDumpURL	string	no	Version string of the request/response elements 'GetCoreDumpURL'.
GetDbTransferState	string	no	Version string of the request/response elements 'GetDbTransferState'.
GetDECTAuthCode	string	no	Version string of the request/response elements 'GetDECTAuthCode'.
GetDECTEncryption	string	no	Version string of the request/response elements 'GetDECTEncryption'.
GetDECTPagingAreaSize	string	no	Version string of the request/response elements 'GetDECTPagingAreaSize'.
GetDECTPowerLimit	string	no	Version string of the request/response elements 'GetDECTPowerLimit'.
GetDECTPpSettings	string	no	Version string of the request/response elements 'GetDECTPpSettings'.
GetDECTRegDomain	string	no	Version string of the request/response elements 'GetDECTRegDomain'.
GetDECTSubscriptionMode	string	no	Version string of the request/response elements 'GetDECTSubscriptionMode'.
GetDevAutoCreate	string	no	Version string of the request/response

			elements 'GetDevAutoCreate'.
GetDigitTreatment	string	no	Version string of the request/response elements 'GetDigitTreatment'.
GetDigitTreatmentSummary	string	no	Version string of the request/response elements 'GetDigitTreatmentSummary'.
GetDTMF	string	no	Version string of the request/response elements 'GetDTMF'.
GetEULAConfirm	string	no	Version string of the request/response elements 'GetEULAConfirm'.
GetEventLogBuffer	string	no	Version string of the event element 'GetEventLogBuffer'.
GetFACList	string	no	Version string of the request/response elements 'GetFACList'.
GetFACPrefix	string	no	Version string of the request/response elements 'GetFACPrefix'.
GetFile	string	no	Version string of the request/response elements 'GetFile'.
GetFlashMemUsage	string	no	Version string of the request/response elements 'GetFlashMemUsage'.
GetFreeConferenceChannels	string	no	Version string of the request/response elements 'GetFreeConferenceChannels'.
GetG729ChannelsForConference	string	no	Version string of the request/response elements 'GetG729ChannelsForConference'.
GetHealthState	string	no	Version string of the request/response elements 'GetHealthState'.
GetIMA	string	no	Version string of the request/response elements 'GetIMA'.
GetIntercomCallHandlingSIP	string	no	Version string of the event element 'GetIntercomCallHandlingSIP'.
GetLastPPDevAction	string	no	Version string of the request/response elements 'GetLastPPDevAction'.
GetLDAP	string	no	Version string of the request/response elements 'GetLDAP'.
GetLicense	string	no	Version string of the request/response elements 'GetLicense'.
GetLicenseServerList	string	no	Version string of the request/response elements 'GetLicenseServerList'.

GetLicensedCodeclines	string	no	Version string of the request/response elements 'GetLicensedCodeclines'.
GetNetParams	string	no	Version string of the request/response elements 'GetNetParams'.
GetOMMCertificate	string	no	Version string of the request/response elements 'GetOMMCertificate'.
GetOMPURL	string	no	Version string of the request/response elements 'GetOMPURL'.
GetNTPServer	string	no	Version string of the request/response elements 'GetNTPServer'.
GetPARK	string	no	Version string of the request/response elements 'GetPARK'.
GetPBXEnrolment	string	no	Version string of the request/response elements 'GetPBXEnrolment'.
GetPortRangeSIP	string	no	Version string of the event element 'GetPortRangeSIP'.
GetPpProfile	string	no	Version string of the request/response elements 'GetPpProfile'.
GetPPDev	string	no	Version string of the request/response elements 'GetPPDev'.
GetPPDevSummary	string	no	Version string of the request/response elements 'GetPPDevSummary'.
GetPPFirmwareUpdate	string	no	Version string of the request/response elements 'GetPPFirmwareUpdate'.
GetPPFirmwareUpdateOverview	string	no	Version string of the request/response elements 'GetPPFirmwareUpdateOverview'.
GetPPFirmwareUpdateStatus	string	no	Version string of the request/response elements 'GetPPFirmwareUpdateStatus'.
GetPPFirmwareURL	string	no	Version string of the request/response elements 'GetPPFirmwareURL'.
GetPPLoginVariant	string	no	Version string of the request/response elements 'GetPPLoginVariant'.
GetPPState	string	no	Version string of the request/response elements 'GetPPState'.
GetPPUser	string	no	Version string of the request/response elements 'GetPPUser'.
GetPPUserSummary	string	no	Version string of the request/response

			elements 'GetPPUserSummary'.
GetPreserveUserDeviceRelation	string	no	Version string of the event element 'GetPreserveUserDeviceRelation'.
GetPublicKey	string	no	Version string of the request/response elements 'GetPublicKey'.
GetReadyForConferencing	string	no	Version string of the request/response elements 'GetReadyForConferencing'.
GetRegistrationTrafficShaping	string	no	Version string of the request/response elements 'GetRegistrationTrafficShaping'.
GetRemoteAccess	string	no	Version string of the request/response elements 'GetRemoteAccess'.
GetRemoteSystemDump	string	no	Version string of the request/response elements 'GetRemoteSystemDump'.
GetRestrictedSubscriptionDuration	string	no	Version string of the event element 'GetRestrictedSubscriptionDuration'.
GetRFP	string	no	Version string of the request/response elements 'GetRFP'.
GetRFPCapture	string	no	Version string of the request/response elements 'GetRFPCapture'.
GetRFPCaptureList	string	no	Version string of the request/response elements 'GetRFPCaptureList'.
GetRFPIpQuality	string	no	Version string of the request/response elements 'GetRFPIpQuality'.
GetRFPM	string	no	Version string of the request/response elements 'GetRFPM'.
GetRFPMediaStreamQuality	string	no	Version string of the request/response elements 'GetRFPMediaStreamQuality'.
GetRFPStatistic	string	no	Version string of the request/response elements 'GetRFPStatistic'.
GetRFPStatisticConfig	string	no	Version string of the request/response elements 'GetRFPStatisticConfig'.
GetRFPSummary	string	no	Version string of the request/response elements 'GetRFPSummary'.
GetRFPSTync	string	no	Version string of the request/response elements 'GetRFPSTync'.
GetRFPSTyncQuality	string	no	Version string of the request/response elements 'GetRFPSTyncQuality'.

GetRTP	string	no	Version string of the request/response elements 'GetRTP'.
GetSARI	string	no	Version string of the request/response elements 'GetSARI'.
GetSecureSIP	string	no	Version string of the request/response elements 'GetSecureSIP'.
GetSecureOMMCertificateServerImport	string	no	Version string of the request/response elements 'GetSecureOMMCertificateServerImport'.
GetSecurePROVCertificateServerImport	string	no	Version string of the request/response elements 'GetSecurePROVCertificateServerImport'.
GetSecureSIPCertificate	string	no	Version string of the request/response elements 'GetSecureSIPCertificate'.
GetSecureSIPCertificateServerImport	string	no	Version string of the request/response elements 'GetSecureSIPCertificateServerImport'.
GetSite	string	no	Version string of the request/response elements 'GetSite'.
GetSiteSummary	string	no	Version string of the request/response elements 'GetSiteSummary'.
GetSNMP	string	no	Version string of the request/response elements 'GetSNMP'.
GetSoftwareImageURL	string	no	Version string of the request/response elements 'GetSoftwareImageURL'.
GetSpecialBranding	string	no	Version string of the request/response elements 'GetSpecialBranding'.
GetStbState	string	no	Version string of the request/response elements 'GetStbState'.
GetSuplServ	string	no	Version string of the request/response elements 'GetSuplServ'.
GetSyslogServer	string	no	Version string of the request/response elements 'GetSyslogServer'.
GetSysStatisticConfig	string	no	Version string of the request/response elements 'GetSysStatisticConfig'.
GetSysStatisticMinMax	string	no	Version string of the request/response elements 'GetSysStatisticMinMax'.
GetSysStatisticMinMaxRecord	string	no	Version string of the request/response

			elements 'GetSysStatisticMinMaxRecord'.
GetSysStatisticMinMaxSummary	string	no	Version string of the request/response elements 'GetSysStatisticMinMaxSummary'.
GetSysStatisticOccurrence	string	no	Version string of the request/response elements 'GetSysStatisticOccurrence'.
GetSystemCredentials	string	no	Version string of the request/response elements 'GetSystemCredentials'.
GetSystemName	string	no	Version string of the request/response elements 'GetSystemName'.
GetSystemProvUpdTrig	string	no	Version string of the request/response elements 'GetSystemProvUpdTrig'.
GetSysToneScheme	string	no	Version string of the request/response elements 'GetSysToneScheme'.
GetSysVoiceboxNum	string	no	Version string of the request/response elements 'GetSysVoiceboxNum'.
GetTimeZone	string	no	Version string of the request/response elements 'GetTimeZone'.
GetTimeZoneDetails	string	no	Version string of the request/response elements 'GetTimeZoneDetails'.
GetTimeZoneList	string	no	Version string of the request/response elements 'GetTimeZoneList'.
GetUsedConfigURL	string	no	Version string of the request/response elements 'GetUsedConfigURL'.
GetUserDataServer	string	no	Version string of the request/response elements 'GetUserDataServer'.
GetUserDeviceSyncOMM	string	no	Version string of the request/response elements 'GetUserDeviceSyncOMM'.
GetUserMonitoring	string	no	Version string of the request/response elements 'GetUserMonitoring'.
GetVersions	string	no	Version string of the request/response elements 'GetVersions'.
GetVideoDev	string	no	Version string of the request/response elements 'GetVideoDev'.
GetVideoDevLink	string	no	Version string of the request/response elements 'GetVideoDevLink'.
GetVideoDevSummary	string	no	Version string of the event element 'GetVideoDevSummary'.

GetWLANClients	string	no	Version string of the request/response elements 'GetWLANClients'.
GetWLANProfile	string	no	Version string of the request/response elements 'GetWLANProfile'.
GetWLANRegDomain	string	no	Version string of the request/response elements 'GetWLANRegDomain'.
GetWLANRegDomainList	string	no	Version string of the request/response elements 'GetWLANRegDomainList'.
GetXMLApplication	string	no	Version string of the request/response elements 'GetXMLApplication'.
Limits	string	no	Version string of the request/response elements 'Limits'.
ManualDBBackup	string	no	Version string of the request/response elements 'ManualDBBackup'.
ManualDBRestore	string	no	Version string of the request/response elements 'ManualDBRestore'.
ManualUserDataImport	string	no	Version string of the event element 'ManualUserDataImport'.
Open	string	no	Version string of the request/response elements 'Open'.
PARKFromServer	string	no	Version string of the request/response elements 'PARKFromServer'.
Ping	string	no	Version string of the request/response elements 'Ping'.
PutFile	string	no	Version string of the request/response elements 'PutFile'.
RequestPositionInfo	string	no	Version string of the request/response elements 'RequestPositionInfo'.
ResetRFPMediaStreamQuality	string	no	Version string of the request/response elements 'ResetRFPMediaStreamQuality'.
ResetRFPStatistic	string	no	Version string of the request/response elements 'ResetRFPStatistic'.
ResetSysStatistic	string	no	Version string of the request/response elements 'ResetSysStatistic'.
ResetTimeZoneDetails	string	no	Version string of the request/response elements 'ResetTimeZoneDetails'.
SendMessage	string	no	Version string of the request/response

			elements 'SendMessage'.
SetAccount	string	no	Version string of the request/response elements 'SetAccount'.
SetACLEntry	string	no	Version string of the request/response elements 'SetACLEntry'.
SetAdvancedSIP	string	no	Version string of the request/response elements 'SetAdvancedSIP'.
SetAlarmTrigger	string	no	Version string of the request/response elements 'SetAlarmTrigger'.
SetAutoDBBackup	string	no	Version string of the request/response elements 'SetAutoDBBackup'.
SetBasicSIP	string	no	Version string of the request/response elements 'SetBasicSIP'.
SetBackupSIP	string	no	Version string of the request/response elements 'SetBackupSIP'.
SetBluetoothBeacon	string	no	Version string of the request/response elements 'SetBluetoothBeacon'.
SetBluetoothGlobalSettings	string	no	Version string of the request/response elements 'SetBluetoothGlobalSettings'.
SetBluetoothSensitivity	string	no	Version string of the request/response elements 'SetBluetoothSensitivity'.
SetConferenceRoom	string	no	Version string of the request/response elements 'SetConferenceRoom'.
SetConferenceServerSIP	string	no	Version string of the request/response elements 'SetConferenceServerSIP'.
SetConfigURL	string	no	Version string of the request/response elements 'SetConfigURL'.
SetCoreDumpURL	string	no	Version string of the request/response elements 'SetCoreDumpURL'.
SetDECTAuthCode	string	no	Version string of the request/response elements 'SetDECTAuthCode'.
SetDECTEncryption	string	no	Version string of the request/response elements 'SetDECTEncryption'.
SetDECTPagingAreaSize	string	no	Version string of the request/response elements 'SetDECTPagingAreaSize'.
SetDECTPowerLimit	string	no	Version string of the request/response elements 'SetDECTPowerLimit'.

SetDECTPpSettings	string	no	Version string of the request/response elements 'SetDECTPpSettings'.
SetDECTRegDomain	string	no	Version string of the request/response elements 'SetDECTRegDomain'.
SetDECTSubscriptionMode	string	no	Version string of the request/response elements 'SetDECTSubscriptionMode'.
SetDevAutoCreate	string	no	Version string of the request/response elements 'SetDevAutoCreate'.
SetDigitTreatment	string	no	Version string of the request/response elements 'SetDigitTreatment'.
SetDTMF	string	no	Version string of the request/response elements 'SetDTMF'.
SetEULAConfirm	string	no	Version string of the request/response elements 'SetEULAConfirm'.
SetFAC	string	no	Version string of the request/response elements 'SetFAC'.
SetFACList	string	no	Version string of the request/response elements 'SetFACList'.
SetFACPrefix	string	no	Version string of the request/response elements 'SetFACPrefix'.
SetIIMA	string	no	Version string of the request/response elements 'SetIIMA'.
SetIntercomCallHandlingSIP	string	no	Version string of the event element 'SetIntercomCallHandlingSIP'.
SetLDAP	string	no	Version string of the request/response elements 'SetLDAP'.
SetLicenseServerList	string	no	Version string of the request/response elements 'SetLicenseServerList'.
SetNetParams	string	no	Version string of the request/response elements 'SetNetParams'.
SetOMMCertificate	string	no	Version string of the request/response elements 'SetOMMCertificate'.
SetOMPURL	string	no	Version string of the request/response elements 'SetOMPURL'.
SetNTPServer	string	no	Version string of the request/response elements 'SetNTPServer'.

SetPARK	string	no	Version string of the request/response elements 'SetPARK'.
SetPBXEnrolment	string	no	Version string of the request/response elements 'SetPBXEnrolment'.
SetPortRangeSIP	string	no	Version string of the event element 'SetPortRangeSIP'.
SetPpProfile	string	no	Version string of the request/response elements 'SetPpProfile'.
SetPPDev	string	no	Version string of the request/response elements 'SetPPDev'.
SetPPFirmwareUpdate	string	no	Version string of the request/response elements 'SetPPFirmwareUpdate'.
SetPPFirmwareURL	string	no	Version string of the request/response elements 'SetPPFirmwareURL'.
SetPPLLoginVariant	string	no	Version string of the request/response elements 'SetPPLLoginVariant'.
SetPPUser	string	no	Version string of the request/response elements 'SetPPUser'.
SetPPUserDevRelation	string	no	Version string of the request/response elements 'SetPPUserDevRelation'.
SetPPUserTracking	string	no	Version string of the request/response elements 'SetPPUserTracking'.
SetPreserveUserDeviceRelation	string	no	Version string of the event element 'SetPreserveUserDeviceRelation'.
SetRegistrationTrafficShaping	string	no	Version string of the request/response elements 'SetRegistrationTrafficShaping'.
SetRemoteAccess	string	no	Version string of the request/response elements 'SetRemoteAccess'.
SetRemoteSystemDump	string	no	Version string of the request/response elements 'SetRemoteSystemDump'.
SetRestrictedSubscriptionDuration	string	no	Version string of the event element 'SetRestrictedSubscriptionDuration'.
SetRFP	string	no	Version string of the request/response elements 'SetRFP'.
SetRFPCapture	string	no	Version string of the request/response elements 'SetRFPCapture'.
SetRFPM	string	no	Version string of the request/response

			elements 'SetRFPM'.
SetRTP	string	no	Version string of the request/response elements 'SetRTP'.
SetRTPConferenceStreamChg	string	no	Version string of the request/response elements 'SetRTPConferenceStreamChg'.
SetSARI	string	no	Version string of the request/response elements 'SetSARI'.
SetSecureSIP	string	no	Version string of the request/response elements 'SetSecureSIP'.
SetSecureOMMCertificateServerImport	string	no	Version string of the request/response elements 'SetSecureOMMCertificateServerImport'.
SetSecurePROVCertificateServerImport	string	no	Version string of the request/response elements 'SetSecurePROVCertificateServerImport'.
SetSecureSIPCertificate	string	no	Version string of the request/response elements 'SetSecureSIPCertificate'.
SetSecureSIPCertificateServerImport	string	no	Version string of the request/response elements 'SetSecureSIPCertificateServerImport'.
SetSite	string	no	Version string of the request/response elements 'SetSite'.
SetSNMP	string	no	Version string of the request/response elements 'SetSNMP'.
SetSoftwareImageURL	string	no	Version string of the request/response elements 'SetSoftwareImageURL'.
SetSpecialBranding	string	no	Version string of the request/response elements 'SetSpecialBranding'.
SetSuplServ	string	no	Version string of the request/response elements 'SetSuplServ'.
SetSyslogServer	string	no	Version string of the request/response elements 'SetSyslogServer'.
SetSystemCredentials	string	no	Version string of the request/response elements 'SetSystemCredentials'.
SetSystemName	string	no	Version string of the request/response elements 'SetSystemName'.
SetSystemProvUpdTrig	string	no	Version string of the request/response elements 'SetSystemProvUpdTrig'.

SetSysToneScheme	string	no	Version string of the request/response elements 'SetSysToneScheme'.
SetSysVoiceboxNum	string	no	Version string of the request/response elements 'SetSysVoiceboxNum'.
SetTimeZone	string	no	Version string of the request/response elements 'SetTimeZone'.
SetTimeZoneDetails	string	no	Version string of the request/response elements 'SetTimeZoneDetails'.
SetUserDataServer	string	no	Version string of the request/response elements 'SetUserDataServer'.
SetUserDeviceSyncOMM	string	no	Version string of the request/response elements 'SetUserDeviceSyncOMM'.
SetUserMonitoring	string	no	Version string of the request/response elements 'SetUserMonitoring'.
SetVideoDev	string	no	Version string of the request/response elements 'SetVideoDev'.
SetWLANProfile	string	no	Version string of the request/response elements 'SetWLANProfile'.
SetWLANRegDomain	string	no	Version string of the request/response elements 'SetWLANRegDomain'.
SetXMLApplication	string	no	Version string of the request/response elements 'SetXMLApplication'.
SoftwareUpate	string	no	Version string of the request/response elements 'SoftwareUpate'.
Subscribe	string	no	Version string of the request/response elements 'Subscribe'.
SystemRestart	string	no	Version string of the request/response elements 'SystemRestart'.

For *errCode* values see 3.3.

4.1.3 Limits

This request is sent from the client to the OMM to request the limits of the OMM.

Following fields are defined for the request *Limits* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *LimitsResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
axiClients	integer	yes	Maximal number of supported AXI clients.
codec	integer	yes	Maximal number of supported codecs.
ldapServer	integer	yes	Maximal number of supported LDAP servers.
licLatency	integer	yes	Grace period timeout in seconds.
ppnNum	integer	yes	Maximal number of supported DECT phones.
records	integer	yes	Maximal number of records supported in one XML element.
rfpNum	integer	yes	Maximal number of supported RFPs.
site	integer	yes	Maximal number of supported sites.
ssidWlan	integer	yes	Maximal number of supported WLAN SSIDs.
ssidWlanKey	integer	yes	Maximal number of supported WLAN keys of an SSID.
ppProfileNum	integer	yes	Maximal number of supported DECT phone profiles.
certificateNum	integer	yes	Maximal number of supported SIP (trusted or local) certificates.
ppVideoDevNum	integer	yes	Maximal number of supported video devices for a DECT phone.
trigger	integer	yes	Maximal number of supported alarm triggers.
urlLen	integer	yes	Maximal supported length of an URL.
userId	integer	yes	Maximal number of user account Ids.
dnsServer	integer	yes	Maximal number of supported RFPM DNS servers.
ntpServer	integer	yes	Maximal number of supported RFPM NTP servers.
wlanProfiles	integer	no	Maximal number of supported WLAN profiles.
wlanClients	integer	no	Maximal number of supported WLAN clients.
wlanMacFilter	integer	yes	Maximal number of supported WLAN MAC filters entries.
xmlBuiltInAppl	integer	yes	Maximal number of supported built-in XML applications.
bluetoothBeacons	integer	yes	Maximal number of supported Bluetooth beacons.
bluetoothNeighbours	integer	yes	Maximal number of supported Bluetooth beacon neighbors.
bluetoothRssiValues	integer	yes	Maximal number of supported Bluetooth RSSI values for a

			Bluetooth client.
conferenceRooms	integer	yes	Maximal number of supported conference rooms.
xmlDynAppl	integer	yes	Maximal number of supported dynamic/configurable XML applications.
xmlCorpDirAppl	integer	yes	Maximal number of supported corporate directory XML applications contained in dynamic/configurable XML application entries.
limitRTT1	yes	yes	IP quality limit 1 for round trip time (rtt) in ms. Interval 1: 0 ms < <i>limitRTT1</i> ms
limitRTT2	yes	yes	IP quality limit 2 for round trip time (rtt) in ms. Interval 2: <i>limitRTT1</i> ms < <i>limitRTT2</i> ms
limitRTT3	yes	yes	IP quality limit 3 for round trip time (rtt) in ms. Interval 3: <i>limitRTT2</i> ms < <i>limitRTT3</i> ms
limitRTT4	yes	yes	IP quality limit 4 for round trip time (rtt) in ms. Interval 4: <i>limitRTT3</i> ms < <i>limitRTT4</i> ms Interval 5: >= <i>limitRTT4</i> ms

For *errCode* values see 3.3.

4.1.4 Subscribe

This request can be sent to OM AXI to change events the client wants to be notified of.

If the underlying TCP link gets closed by whatever reason, all notification subscriptions end automatically.

The subscription mechanism remembers a flag for each possible event and for each possible element (e. g. DECT phone). The command “*on*” sets the corresponding flag in the OM AXI implementation for the given events. It can be reset by using “*off*”.

Following fields are defined for the request *Subscribe* in addition to the common attributes:

Name	Type	Mandatory	Description
e	sequence of <i>SubscribeCmdType</i>	no	Up to 20 event commands to be executed at once atomically.

The following table shows the type *SubscribeCmdType*:

Name	Type	Mandatory	Description
cmd	enumeration	yes	“On” to set the subscription flag, “Off” to reset it.
eventType	<i>EventType</i>	yes	The event type.
ppn	integer	no	The subscription command is valid for this DECT phone device. If it is -1, the command automatically refers to all items. If it is missing, no items are included.
uid	integer	no	The subscription command is valid for this DECT phone user. If it is -1, the command automatically refers to all items. If it is

			missing, no items are included.
rfpld	integer	no	The subscription command is valid for this RFP. If it is -1, the command automatically refers to all RFPs. If it is missing, no RFPs are included.
omm	integer	no	The subscription command is valid for the OMM if this field has the value -1. If it is missing, the OMM is not included.
trigger	string	no	Alarm triggers the client is interested in. If it is an empty String (""), the commands automatically refers to all alarm triggers. If it is missing, no triggers are included. Wildcards like "LOC-*" are also allowed.
scheme	string	no	Address scheme the client is interested in (without ':'). If it is an empty String (""), the commands automatically refers to all address schemes. If it is missing, no address scheme is included.

For some events it is possible to limit the command to one or all items of a certain type , e. g. to one DECT phone. Which of the filters are actually available for a certain event is documented in the related chapters. These optional fields are ignored for events which do not support this filter type.

Important: If several filters are given, they are linked with a logical *or*. Example: Subscribing for *EventAlarmTrigger* with filters *ppn="7"* and *trigger="mandown"* causes all alarms coming from PPN 7 **or** with the trigger *"mandown"* to be sent. When a subscription for all ppn is deleted (*ppn="-1" cmd="off"*) all previous set subscriptions for ppn are delete not even the *ppn="-1"* subscription.

This mechanism can be used to build advanced filters from the client. For example, to subscribe notifications for event X for all DECT phones but ppn 77, one could set the flags for all DECT phones at once and then reset the flag for X with ppn 77. This is shown as an example below to make it clearer. All commands contained in one *Subscribe* request are processed atomically in the OMM.

The reply to this request is called *SubscribeResp*. Note that this is not the notification itself but the confirmation for *Subscribe*. If an error is reported, none of the commands was executed. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
eventType	EventType	No	The first event type which caused EInval or EPerm

Possible values for EventType and usage of additional parameters:

EventType	ppn	uid	rfpld	omm	trigger	scheme
AccountCnf						
AccountSummary						
ACLCnf						
AlarmCallProgress	x				x	
AlarmTrigger	x			x	x	
AlarmTriggerCnf						

AutoDBCnf						
BluetoothCnf						
BluetoothBeaconSumCnf						
BluetoothBeaconSumState						
Conference						
ConferenceCnf						
DECTSubscriptionMode						
DigitTreatmentCnf						
EventLog						
FACCnf						
LicensedCodeclines						
LicensedFile						
Locating	x					
LocatingBluetooth						
MessageConfirmation	x					
MessageProgress	x					
MessageQueueEmpty	x					
MessageSend	x					x
ParkServerResult						
PositionTrack	x					
PPHCM						
PPDevCnf	x					
PPFirmwareUpdateCnf						
PPFirmwareUpdateOverview						
PPFirmwareUpdateState	x					
PPFirmwareState						
PPState	x					

PPBtState	x					
PPCallState	x					
PPCallStateRinging	x					
PPCallStateCalling	x					
PPCallStatePaging	x					
PPCallStateConnected	x					
PPCallStateIdle	x					
PPCallStateNone	x					
PPDevSummary						
PPDevState	x					
PPDevStateOnHook	x					
PPDevStateSilentCharging	x					
PPDevStateBatteryLevel	x					
PPDevStateSwVersion	x					
PPSipState	x					
PPSipStateRegEvent	x					
PPSipStateEndEvent	x					
PPSipStateNotifyEvent	x					
PPSipStateRegistered	x					
PPSipStateServerType	x					
PPSipStateServerAddr	x					
PPSipStateServerPort	x					
PPTransaction	x					
PPTransactionLinkEstablish	x					
PPTransactionRelease	x					
PPTransactionEstablish	x					
PPTransactionPPNotFound	x					

PPTransactionPagingStarted	x					
PPTransactionReleaseFromPP	x					
PPTransactionPPSetupRejected	x					
PPTransactionComsRelease	x					
PPTransactionComsEstablish	x					
PPTransactionSsFac	x					
PPTransactionSsRelease	x					
PPTransactionSsEstablish	x					
PPTransactionConnHandover	x					
PPTransactionLocReg	x					
PPTransactionDetach	x					
PPUserCnf		x				
PPUserSummary						
RFPCnf			x			
RFPCConnectAttempt						
RFPDetails			x			
RFPipQuality			x			
RFPMsQuality			x			
RFPState			x			
RFPsyncQuality			x			
RFPsummary						
RFPsync			x			
Roaming						
SiteCnf						
SiteSummary						
SystemCnf						
SystemState						

VideoDev						
VideoDevSumCnf						
VideoDevSumState						
WLANClient						
WLANProfileCnf						
UserDataImport						

Possible values for *errCode*:

Value	Description
EInval	Unknown event type
ENoMem	Too many <i>trigger</i> or <i>scheme</i> subscription commands tried to set. Maximal 25 of those are supported for each client.

For further *errCode* values see 3.3.

The client needs one of these permissions to use this request.

Permission	Description
AllCnfRead	Can subscribe to all configuration changes and to <i>SystemState</i> , <i>RFPState</i> , <i>DECTSubscriptionMode</i> , <i>RFPDetails</i> , <i>RFPConnectAttempt</i> , <i>WLANClient</i> , <i>AlarmTrigger</i> , <i>AlarmCallProgress</i> , <i>PositionTrack</i> , <i>PPFirmwareUpdateCnf</i> , <i>PPFirmwareUpdateOverview</i> , <i>PPFirmwareUpdateState</i> , <i>PPFirmwareState</i> , <i>FACCnf</i> , <i>BluetoothCnf</i> , <i>BluetoothBeaconSumCnf</i> , <i>BluetoothBeaconSumState</i> , <i>Conference</i> , <i>LicensedCodeLines</i> , <i>LicenseFile</i> , <i>VideoDev</i> , <i>VideoDevSumCnf</i> , <i>VideoDevSumState</i> , <i>PPTransaction</i> , <i>PPTransactionLinkEstablish</i> , <i>PPTransactionRelease</i> , <i>PPTransactionEstablish</i> , <i>PPTransactionPPNotFound</i> , <i>PPTransactionPagingStarted</i> , <i>PPTransactionReleaseFromPP</i> , <i>PPTransactionPPSetupRejected</i> , <i>PPTransactionComsRelease</i> , <i>PPTransactionComsEstablish</i> , <i>PPTransactionSsFac</i> , <i>PPTransactionSsRelease</i> , <i>PPTransactionSsEstablish</i> , <i>PPTransactionConnHandover</i> , <i>PPTransactionLocReg</i> , <i>PPTransactionDetach</i> , <i>AccountSummary</i> , <i>PPDevSummary</i> , <i>PPUserSummary</i> , <i>RFPsSummary</i> , <i>SiteSummary</i> , <i>RFPipQuality</i> , <i>RFPmsQuality</i> , <i>RFPsyncQuality</i> , <i>PCHCM</i>
Messaging or InfoMessaging	Can subscribe to <i>MessageSend</i> , <i>MessageProgress</i> , <i>MessageConfirmation</i> , <i>MessageQueueEmpty</i>
Locating	Can subscribe to <i>Locating</i> , <i>PositionTrack</i> , <i>PPTransaction</i> , <i>PPTransactionLinkEstablish</i> , <i>PPTransactionRelease</i> , <i>PPTransactionEstablish</i> , <i>PPTransactionPPNotFound</i> , <i>PPTransactionPagingStarted</i> , <i>PPTransactionReleaseFromPP</i> , <i>PPTransactionPPSetupRejected</i> , <i>PPTransactionComsRelease</i> , <i>PPTransactionComsEstablish</i> , <i>PPTransactionSsFac</i> , <i>PPTransactionSsRelease</i> , <i>PPTransactionSsEstablish</i> , <i>PPTransactionConnHandover</i> , <i>PPTransactionLocReg</i> , <i>PPTransactionDetach</i>
Monitoring	Can subscribe to <i>PositionTrack</i> , <i>PPTransaction</i> , <i>RFPsync</i> , <i>Conference</i> , <i>PPState</i> , <i>PPBtState</i> , <i>PPCallState</i> , <i>PPDevState</i> , <i>PPSipState</i> , <i>PPCallStateRinging</i> , <i>PPCallStateCalling</i> , <i>PPCallStatePaging</i> , <i>PPCallStateConnected</i> , <i>PPCallStateIdle</i> , <i>PPCallStateNone</i> , <i>PPDevStateOnHook</i> , <i>PPDevStateSilentCharging</i> ,

	<i>PPDevStateBatteryLevel, PPDevStateSwVersion, PPSipStateRegEvent, PPSipStateEndEvent, PPSipStateNotifyEvent, PPSipStateRegistered, PPSipStateServerType, PPSipStateServerAddr, PPSipStateServerPort, PPTransactionLinkEstablish, PPTransactionRelease, PPTransactionEstablish, PPTransactionPPNotFound, PPTransactionPagingStarted, PPTransactionReleaseFromPP, PPTransactionPPSetupRejected, PPTransactionComsRelease, PPTransactionComsEstablish, PPTransactionSsFac, PPTransactionSsRelease, PPTransactionSsEstablish, PPTransactionConnHandover, PPTransactionLocReg, PPTransactionDetach</i>
--	---

4.1.5 Ping

With this request a client can keep the TCP link alive. If the request is sent before the 5 minutes timeout expires, the link is kept open.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
timeStamp	long integer	no	The client requests with the time stamp (in seconds since 1970/1/1) the current time stamp from the AXI server (OMM).

The reply is an object called *PingResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
timeStamp	long integer	no	Current time stamp (in seconds since 1970/1/1) response by the AXI server (OMM) when the client requires the time stamp to verify whether the current time is correctly used.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: (any)

4.1.6 GetStbState

With this requests the client can ask for information about the current OMM standby state.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The response to this request does also contain information about the OMM standby feature.

The reply is an object called *GetStbStateResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
ommStbState	enumeration	yes	State of the OMM standby feature. Refer to <i>Open</i> (chapter 4.1.1) for more information.
ommStbAddr	string	no	In case of OMM standby is active: IP address of the other OMM

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request. *AllCnfRead*

4.1.7 EventStbStateChange

This event is sent by OM AXI when the OMM standby state has changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemState*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
ommStbState	enumeration	yes	State of the OMM standby feature. Refer to <i>Open</i> (chapter 4.1.1) for more information.
ommStbAddr	string	no	In case of OMM standby is active: IP address of the other OMM

Available filters: (none)

4.1.8 EventPermissionChange

This event is sent by OM AXI when the OMM permissions have changed (e. g. change license conditions). Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemState*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
permission	sequence of PermissionType	no	All Permissions newly given to this client. They may depend on the user account used for authentication and on the current license state.

Available filters: (none)

4.1.9 GetPublicKey

With this request the client can obtain the public RSA key to be used to encrypt certain attributes. Refer to chapter 2.4 for details.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetPublicKeyResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
modulus	string	yes	Hexadecimal encoded public modulus n
exponent	string	yes	Hexadecimal encoded public exponent e

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request. *AllCnfRead*

4.2 License Management

The license configuration can also be done by downloading the complete licenses file from the license server to the OMM (see 4.74.2). For additional information about licensing see the OMM installation guide and chapter 5.5.

4.2.1 LicenseType

This type contains all license RFP data fields. It is used in license the event and license responses defined in this chapter.

Name	Type	Description
key	string	License key string
number	integer	Up to this number the license is valid depending on the license type: OMM system license: Number of RFPs to be used OMM locating license: Number of DECT phones allowed users to locate other users
systemLicenseVersion	string	OMM system license: „x.y”, specifies the major release number “x” and the minor release number “y” of the OMM release the license is valid for. Only used in the <i>sysLicense</i> element.
messagingLicenseRcvMsgs	boolean	„1” or “true”, if the license is valid to receive messages at the DECT phones. Only used in the <i>msgLicense</i> element.
locatingLicense	boolean	„1” or “true”, if the license is valid to run the licensed external location application. Only used in the <i>locLicense</i> element.

4.2.2 LicenseRFPType

This type contains all license RFP data fields. It is used in license the event and license responses defined in this chapter.

Name	Type	Description
id	integer	Unique RFP identifier of the license RFP
ethAddr	string	Ethernet address of the license RFP, format “00:11:22:aa:bb:cc”
connected	boolean	„1” or “true”, if the license RFP is connected

4.2.3 GetLicense

With this request the client can query the license configuration of the OMM.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Description
		(none)

The reply is an object called *GetLicenseResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
type	enumeration	yes	Type of license, values are one of "small" (<i>own PARK, no activation for maximal 5 RFPs</i>), "standard" (<i>own PARK and license for all RFPs</i>)
state	enumeration	yes	State of any license violation. Values are one of "noLicense" (<i>no valid OMM system license set</i>), "inactiveLicense" (<i>less than 2 license RFPs connected</i>), "inactiveLicenseBlocked" (<i>configuration exceeds license limitation, e. g. too many RFPs configured</i>), "noRedundancyLicense" (<i>only 2 license RFPs are connected</i>), "noRedundancyLicenseBlocked" (<i>configuration exceeds license limitation</i>), "activeLicense" (<i>all is fine</i>) or "activeLicenseBlocked" (<i>configuration exceeds license limitation</i>). <i>In the license non violated state "noLicense" or "activeLicense" the "latency"/"grace period" counter (see below) counts up, otherwise it counts down until to the maximum value of 43200 minutes (30 days).</i>
violation	sequence of enumeration	yes	License violation reasons, when the state above is not "noLicense" and not "activeLicense". Each element of the sequence is one of "noViolation" (<i>no license violation</i>), "noLicense" (<i>no OMM system license configured</i>), "numRFPs" (<i>too many RFPs configured</i>), "swRelease" (<i>license not valid for this software release</i>) or "numLocatables" (<i>too many users configured to locate</i>)
latency	integer	yes	Grace period time value in minutes, how long the system will work with a violated license. For an activated and valid license the value has a maximum value of <i>43200 minutes (30 days)</i> . This value counts down when the license is violated and counts up when the license is OK again.
park	string	no	PARK for the license keys.
licenseRfp	sequence of LicenseRFPTy pe	no	Up to three data sets for license RFPs
sysLicense	LicenseType	no	OMM system license.
msgLicense	LicenseType	no	OMM messaging license.
locLicense	LicenseType	no	OMM locating license.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.2.4 EventLicenseCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.44.1.2 on how to set up a notification session.

To get this event the client has to subscribe to *SystemState*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
type	enumeration	no	Type of license, values are one of "small" (<i>own PARK, no activation for maximal 5 RFPs</i>), "standard" (<i>own PARK and license for all RFPs</i>)
state	enumeration	no	State of any license violation. Values are one of "noLicense" (<i>no valid OMM system license set</i>), "inactiveLicense" (<i>less than 2 license RFPs visible</i>), "inactiveLicenseBlocked" (<i>configuration exceeds license limitation, e. g. too many RFPs configured</i>), "noRedundancyLicense" (<i>only 2 license RFPs are connected</i>), "noRedundancyLicenseBlocked" (<i>configuration exceeds license limitation</i>), "activeLicense" (<i>all is fine</i>) or "activeLicenseBlocked" (<i>configuration exceeds license limitation</i>). <i>In the license non violated state "noLicense" or "activeLicense" the "latency"/"grace period" counter (see below) counts up, otherwise it counts down until to the maximum value of 43200 minutes (30 days).</i>
violation	enumeration	no	License violation reasons, when the state above is not "noLicense" and not "activeLicense". Each element of the sequence is one of "noViolation" (<i>no license violation</i>), "noLicense" (<i>no OMM system license configured</i>), "numRFPs" (<i>too many RFPs configured</i>), "swRelease" (<i>license not valid for this software release</i>) or "numLocatables" (<i>too many DECT phones configured to locate</i>)
latency	integer	no	Grace period time value in minutes, how long the system will work with a violated license. For an activated and valid license the value has a maximum value of <i>43200 minutes (30 days)</i> . This value counts down when the license is violated and counts up when the license is OK again.
park	string	no	PARK for the license keys.
licenseRFP	sequence of LicenseRFPTy pe	no	Up to three data sets for license RFPs. This license RFP list is always complete in the event, even if there is no change for a license RFP. This enables the possibility to determine whether an RFP was deleted or was unchanged.
sysLicense	LicenseType	no	OMM system license.

msgLicense	LicenseType	no	OMM messaging license.
locLicense	LicenseType	no	OMM locating license.

Available filters: (none)

4.2.5 GetLicensedCodecLines

With this request the client can query the number of licensed G.729 call connections.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Description
		(none)

The reply is an object called *GetLicensedCodecLinesResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
numberOfLines	integer	yes	Number of currently used licensed G.729 call connections.

Available filters: (none)

4.2.6 EventLicensedCodecLines

This event is sent by OM AXI when the number of licensed G.729 call connections has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

This event is only supported when the "OpenResp" capability parameter *haveLicensingCodec* is set.

To get this event the client has to subscribe to *LicensedCodecLines*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
numberOfLines	integer	yes	Number of currently used licensed G.729 call connections.

Available filters: (none)

4.3 License Server Configuration And Maintenance

4.3.1 LicenseServerType

This type contains all license server data fields. It is used in the license client/server type defined in this chapter.

Name	Type	Description
server	string	IP address or domain name of the license server

port	integer	Port of the license server
serverStandby	string	IP address or domain name of a standby license server for this license server
portStandby	integer	Port for the standby license server

4.3.2 LicenseServerListType

This type contains all license client/server data fields. It is used in the license client/server event and license client/server responses defined in this chapter.

Name	Type	Description
serverActive	boolean	„1” or “true”, if the OMM license server is active
installationId	integer	Unique ID which identifies the OMM system to be licensed.
licenseServer1	LicenseServerType	First license server configuration
licenseServer2	LicenseServerType	Second license server configuration
licenseServer3	LicenseServerType	Third license server configuration

4.3.3 GetLicenseServerList

With this request the client can query the license client/server configuration of the OMM.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Description
		(none)

The reply is an object called *GetLicenseServerListResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
licenseServerList	LicenseServerListType	yes	License client/server configuration data configured

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.3.4 SetLicenseServerList

The client can send this request to OM AXI to change license client/server data set. Additionally the attributes which have to be changed must be filled in by the client.

Following element is defined for the request *SetLicenseServerList* in addition to the common attributes:

Name	Type	Mandatory	Description
licenseServerList	LicenseServerListType	yes	License client/server configuration data to set

If the change was successful, the reply contains all attributes which *GetLicenseServerList* would return. If the change fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetLicenseServerListResp*.

It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
licenseServerList	LicenseServerListType	yes	All set License client/server configuration data

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.3.5 EventLicenseServerListCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
licenseServerList	LicenseServerListType	yes	Changed License client/server configuration data

Available filters: (none)

4.3.6 EventLicenseFile

This event is sent by OM AXI when a new license file or a new license data file (used by the integrated license server) is stored in the OMM. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *LicenseFile*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
type	enumeration	yes	Type of license file that changes, values are one of "LicenseFile" (a new license has been configured), "LicenseDataFile" (The licesnse server stored a new license data file),

Available filters: (none)

4.4 Account Configuration And Maintenance

With the requests and events described in this chapter the user accounts can be configured. These accounts are used e. g. for the *Open* request.

The account with id 0 plays a special role: This is the initial administrator account. This one cannot be deleted. The permissions of this cannot be changed.

4.4.1 AccountType

This type contains all data fields of an account. It is used in different requests and responses defined in this document.

Name	Type	Description
id	integer	Account ID, numbering starts at 0, -1 is invalid.
username	string	User name
password	string	Password, encrypted with public key.
oldPassword	string	Old password, encrypted with public key, for verification. Only needed when password should be changed.
permission	sequence of PermissionType	Permissions for this account.
active	boolean	If true account is active. Optional set/create parameter. Default value=false.
aging	enumeration	Selected type of password aging (timeout or number of logins). On of "none", "time3Months", "time6Months", "count50Logins", "count100Logins". Optional set/create parameter. Default value=none.
expire	integer	Remaining time or logins until password expires. Only a get/event parameter.
state	string	One of <i>dontCare</i> , <i>changed</i> , <i>notChanged</i> , <i>expired</i> .

4.4.2 GetAccountSummary

With this request the client can query common information about the accounts configured.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to this request is an object called *GetAccountSummaryResp*. It contains following fields in addition to the common attributes:

Name	Type	Description
nRecords	integer	The total number of accounts set up.
idFirst	integer	The ID of the first account set up.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.4.3 GetAccount

With this request the client can acquire one or more account data sets from the OMM. The request contains an *id* from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

These IDs usually start at 0, but possibly not all numbers are assigned, i.e. the numbering may not be contiguous.

The responses are ordered by *id*, starting with the smallest.

If the client asked for a certain number of records and the given *id* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given *id* has to match to a record exactly. If there is a record with this *id*, this one is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetAccount* (*id* = "0", *maxRecords* = "20"). If the response contains 20 records, take the last ID plus one and send another *GetAccount*. Repeat this until you get less than 20 records back.

Following fields are defined for the request *GetAccount* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	First ID of accounts to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to this request is an object called *GetAccountResp*. It contains following element in addition to the common attributes:

Name	Type	Description
account	sequence of AccountType	List of accounts

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.4.4 CreateAccount

The client can send this request to OM AXI to create a new account data set.

If there is no *id* assigned by the client or if it is -1, the OMM will choose one. If the client assigns an *id*, it must not exist yet.

Following element is defined for the request *CreateAccount* in addition to the common attributes:

Name	Type	Mandatory	Description
account	AccountType	yes	Data of account to create.

An account must always have a *name*, and a *password*.

If the creation was successful, the reply contains all attributes which *GetAccount* would return. Some data fields may be changed or set by the OMM. If the creation fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreateAccountResp*.

It contains following element in addition to the common attributes:

Name	Type	Description
account	AccountType	Created account

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.4.5 DeleteAccount

With this request the client can delete a site record in the OMM. The key is the *id*.

Following fields are defined for the request *DeleteAccount* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	ID of account to be deleted

The initial administrator account with *id* 0 cannot be deleted.

The reply to this request is an object called *DeleteAccountResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.4.6 SetAccount

The client can send this request to OM AXI to change an account data set. The *id* has to be filled in by the client to identify the record to be changed. Additionally the attributes which have to be changed must be filled in by the client.

Following element is defined for the request *SetAccount* in addition to the common attributes:

Name	Type	Mandatory	Description
account	AccountType	yes	Data of account to change.

If the change was successful, the reply contains all attributes which *GetAccount* would return. Some data fields may be changed or set by the OMM. If the change fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The permissions of the initial administrator account with *id* 0 cannot be changed.

The reply to this request is called *SetAccountResp*.

It contains following element in addition to the common attributes:

Name	Type	Description
account	AccountType	Changed account

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.4.7 EventAccountCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *AccountCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with only an *id* in the *account* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
account	AccountType	yes	Changed attributes
deleted	boolean, default false		„1” or “true”, if this data set has been deleted

Available filters: (none)

4.4.8 EventAccountSummary

This event is sent by OM AXI when the number of accounts has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf* or to *AccountSummary*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
nRecords	integer	yes	The total number of accounts set up.
idFirst	integer	yes	The ID of the first account set up.

Available filters: (none)

4.5 OMM Software Configuration And Management

4.5.1 SoftwareUpdate

With this request the client can activate an OMM software update. In a controlled action all RFPs are forced to request for new software to load.

Following fields are defined for the request *SoftwareUpdate* in addition to the common attributes:

Name	Type	Mandatory	Description
timedUpdate	boolean	no	„1” or “true”, if the software update should be done at a scheduled time
allAtOnce	boolean	no	„1” or “true”, if the software update should be done for all RFPs at a time. If not set the OMM controls the software update

			process for all RFPs one after another within a DECT cluster.
hour	integer	no	Hour time value of the software update point of time.
minute	integer	no	Minute time value of the software update point of time.

Note: In case of no given attribute it is assumed to force an RFP update process (looking for new software to load) for all RFPs at once.

The reply is an object called *SoftwareUpdateResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
timedUpdate	boolean	yes, if any configuration changes no, if an RFP update process is done	„1” or “true”, if the software update should be done at a scheduled time
allAtOnce	boolean	yes, if any configuration changes no, if an RFP update process is done	„1” or “true”, if the software update should be done for all RFPs at a time. If not set the OMM controls the software update process for all RFPs one after another within a DECT cluster.
hour	integer	yes, if any configuration changes no, if an RFP update process is done	Hour time value of the software update point of time.
minute	integer	yes, if any configuration changes no, if an RFP update process is done	Minute time value of the software update point of time.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.5.2 GetSoftwareUpdate

With this request the client can query the data for software update.

Following fields are defined for the request *GetSoftwareUpdate* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetSoftwareUpdateResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
timedUpdate	boolean	yes	„1” or “true”, if the software update should be done at a scheduled time
allAtOnce	boolean	yes	„1” or “true”, if the software update should be done for all RFPs at a time. If not set the OMM controls the software update process for all RFPs one after another within a DECT cluster.
hour	integer	yes	Hour time value of the software update point of time.
minute	integer	yes	Minute time value of the software update point of time.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.5.3 EventSoftwareUpdateCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
timedUpdate	boolean	no	„1” or “true”, if the software update should be done at a scheduled time
allAtOnce	boolean	no	„1” or “true”, if the software update should be done for all RFPs at a time. If not set the OMM controls the software update process for all RFPs one after another within a DECT cluster.
hour	integer	no	Hour time value of the software update point of time.
minute	integer	no	Minute time value of the software update point of time.

Available filters: (none)

4.5.4 SystemRestart

This request is sent from the client to the OMM.

With this request the client can initiate an OMM system restart.

Following fields are defined for the request *SystemRestart* in addition to the common attributes:

Name	Type	Mandatory	Description
resetDB	boolean	yes	„1” or “true”, if the OMM DB and configuration files shall be discarded. OMM DB and configuration files are removed from the RFP incl. the data retrieved from RCS. A local IP configuration will remain unaffected..
resetToFactoryDefaults	boolean	yes	„1” or “true”, if OMM RFP(s) shall be reset to factory defaults. All data are removed from the RFP incl. OMM DB, configuration files and local IP configuration. This attribute is only supported on an RFP running OMM.

The reply to *SystemRestart* is an object called *SystemRestartResp*. It contains following fields in addition to the common attributes:

Name	Type	Description
		(no additional fields defined)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.5.5 OMM Software Image Management

4.5.5.1 GetSoftwareImageURL

With this request the client can query the software image configuration URL settings used to load OMM software files from. Following fields are defined for the request *GetSoftwareImageURL* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to load OMM software files from the external server.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.5.5.2 SetSoftwareImageURL

With this request the client can set the software image URL settings used to load OMM software files from.

Following fields are defined for the request *SetSoftwareImageURL* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to load OMM software files from the external server.

The reply is an object called *SetSoftwareImageURLResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.5.5.3 EventSoftwareImageURLCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to load OMM software files from the external server.

Available filters: (none)

4.5.6 DECT Phone Firmware Management

4.5.6.1 DECT phoneFirmwareUpdateStatusType

This type contains all data fields of DECT phone firmware update status. It is used in different event and responses defined in this chapter.

Name	Type	Mandatory	Description
ppn	integer	yes	PPN of the shown firmware update status.
state	enumeration	yes	Download state of the DECT phone . One of "ready", "pending", "active", "barred", "error", "notReachable", "detached" and "unknown"
cause	enumeration	no	More detailed cause for some states. One of: "none":no more details "busy":DECT phone is currently busy "battery":low battery "barred":SW download disabled locally "noMaster":OMM is not download master "crc": DECT phone detects checksum error "noMem":DECT phones file system full "flash":error while writing SW to flash "system":DECT phone file system error "version":software version mismatch "file":invalid file data "container":error while container expanding
bytes	integer	no	Number of bytes outstanding to load. (only contained in state="active")
version	string	no	Version string of the running software in the DECT phone. (only contained when the DECT phone is connected)

4.5.6.2 GetPPFirmwareUpdate

With this request the client can query the state of DECT phone firmware updates (also known as "download over air").

Following fields are defined for the request *GetPPFirmwareUpdate* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetPPFirmwareUpdateResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1" or "true", if DECT phone firmware updates are enabled
name	string	yes	Filename of the DECT phone firmware.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.5.6.3 SetPPFirmwareUpdate

With this request the client can enable or disable DECT phone firmware updates (also known as "download over air").

Following fields are defined for the request *SetPPFirmwareUpdate* in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, to enable DECT phone firmware updates

The reply is an object called *SetPPFirmwareUpdateResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	no	In case of success: value actually set

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.5.6.4 GetPPFirmwareUpdateOverview

With this request the client can query for the overview status of the DECT phone download status.

Following fields are defined for the request *GetPPFirmwareUpdateOverview* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetPPFirmwareUpdateOverviewResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
state	enumeration	yes	State of the DECT phone download manager. One of “ <i>startup</i> ”, “ <i>disabled</i> ”, “ <i>running</i> ” or “ <i>error</i> ”.
known	integer	no	Number of known DECT phones able to download software. (only contained when state=“ <i>running</i> ”)
ready	integer	no	Number of DECT phones having already the required software version. (only contained when state=“ <i>running</i> ”)
pending	integer	no	Number of DECT phones waiting for a download the new software version. (only contained when state=“ <i>running</i> ”)
active	integer	no	Number of DECT phones downloading the new software version. (only contained when state=“ <i>running</i> ”)
barred	integer	no	Number of DECT phones which are barred for software download. (only contained when state=“ <i>running</i> ”)
errored	integer	no	Number of DECT phones with error during software download. (only contained when state=“ <i>running</i> ”)
notReachable	integer	no	Number of DECT phones which are unreachable. (only contained when state=“ <i>running</i> ”)

detached	integer	no	Number of DECT phones which are detached. (only contained when state="running")
version	string	no	Version string of the required DECT phone software to download. (only contained when state="running")

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.5.6.5 GetPPFirmwareUpdateStatus

With this request the client can acquire one or more DECT phone firmware update/download data sets from the OMM. The request contains a PPN from which starting subsequent DECT phone firmware update/download data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

PPNs usually ordered start at 1, but possibly not all numbers are assigned, i.e. the DECT phone device numbering may not be contiguous.

The responses are by PPN, starting with the smallest.

If the client asked for a certain number records but the given *ppn* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given PPN has to match to a record exactly. If there is a record with this PPN, it is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetPPFirmwareUpdateStatus* (*ppn* = "1", *maxRecords* = "20"). If the response contains 20 records, take the last PPN plus one and send another *GetPPFirmwareUpdateStatus*. Repeat this until you get less than 20 records back.

Following fields are defined for the request *GetPPFirmwareUpdateStatus* in addition to the common attributes:

Name	Type	Mandatory	Description
ppn	integer	yes	First PPN of DECT phone devices to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to *GetPPFirmwareUpdateStatus* is an object called *GetPPFirmwareUpdateStatusResp*. It contains following element in addition to the common attributes:

Name	Type	Description
ppFwSt	sequence of PPFirmwareUpdateStatusType	DECT phone FirmwareUpdateStatus records, if found

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.5.6.6 EventPPFirmwareUpdateCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *PPFirmwareUpdateCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	no	Changed DECT phone firmware update setting
name	string	no	Filename of the DECT phone firmware.

Available filters: (none)

4.5.6.7 EventPPFirmwareUpdateOverview

This event is sent by OM AXI when the DECT phone firmware update/download overview state changes. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *PPFirmwareUpdateOverview*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification are the same as in *GetPPFirmwareUpdateOverview* (see 4.5.6.4).

Available filters: (none)

4.5.6.8 EventPPFirmwareUpdateStatus

This event is sent by OM AXI when the detailed DECT phone firmware update/download status state changes. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *PPFirmwareUpdateState*. It is valid to subscribe for only one DECT phone to get the detailed pp download state. A subscription for a new DECT phone would stop the event notification for the previous DECT phone. Additionally the client can subscribe to *PPFirmwareState* for all DECT phones to get event notifications when only the DECT phone download state changes. Both events are allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in those notifications are as follows.

Name	Type	Description
ppFwSt	PPFirmwareUpdateStatusType	DECT phone FirmwareUpdateStatus

Available filters: (none)

4.5.6.9 GetPPFirmwareURL

With this request the client can query the DECT phone firmware configuration URL settings used to load the DECT phone firmware file from. Following fields are defined for the request *GetPPFirmwareURL* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to load the DECT phone firmware file from the external server.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.5.6.10 SetPPFirmwareURL

With this request the client can set the DECT phone firmware URL settings used to load the DECT phone firmware file from.

Following fields are defined for the request *SetPPFirmwareURL* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to load the DECT phone firmware file from the external server.

The reply is an object called *SetPPFirmwareURLResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.5.6.11 EventPPFirmwareURLCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to load the DECT phone firmware file from the external server.

Available filters: (none)

4.6 System Name Configuration And Maintenance

4.6.1 GetSystemName

With this request the client can query the system name.

Following fields are defined for the request *GetSystemName* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetSystemNameResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
name	string	yes	system name

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.6.2 SetSystemName

With this request the client can set the system name.

Following fields are defined for the request *SetSystemName* in addition to the common attributes:

Name	Type	Mandatory	Description
name	string	yes	system name

The reply is an object called *SetSystemNameResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.6.3 EventSystemNameCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
name	string	yes	Changed system name

Available filters: (none)

4.7 EULA Management

4.7.1 GetEULAConfirm

With this request the client can query if the EULA of this OMM version has been accepted already.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetEULAConfirmResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
confirm	boolean	yes	„1” or “true”, if the EULA has been confirmed already.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.7.2 SetEULAConfirm

With this request the client can confirm the EULA.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
confirm	boolean	yes	„1” or “true”, to confirm the EULA.

When the client sets *confirm* to *false*, i.e. when the user did not agree to the EULA, the application must not continue to use OM AXI.

The reply is an object called *SetEULAConfirmResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.7.3 EventEULAConfirmCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
confirm	boolean	yes	„1” or “true”, to confirm the EULA.

Available filters: (none)

4.8 Remote Access Configuration And Maintenance

4.8.1 GetRemoteAccess

With this request the client can query the configuration of the remote access (i.e. SSH access).

Following fields are defined for the request *GetRemoteAccess* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetRemoteAccessResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, if the remote access is enabled

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.8.2 SetRemoteAccess

With this request the client can set the configuration of the remote access (i.e. SSH access).

Following fields are defined for the request *SetRemoteAccess* in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, to enable the remote access

The reply is an object called *SetRemoteAccessResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.8.3 EventRemoteAccessCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	yes	Changed remote access configuration

Available filters: (none)

4.9 Syslog Server Configuration And Maintenance

4.9.1 GetSyslogServer

This request is sent from the client to the OMM.

With this request the client can query the current settings of the syslog server.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

If the change was successful, the reply contains all attributes which *GetSyslogServer* would return.

The reply to this request is an object called *GetSyslogServerResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, when syslog server is enabled
forward	boolean	no	„1” or “true”, when forwarding of loggingss from OMM running on a PC to syslog server is enabled
ipAddr	string	yes	IP address of syslog server
port	integer	yes	Port number of syslog server

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.9.2 SetSyslogServer

This request is sent from the client to the OMM.

With this request the client can set the syslog server configuration.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, when syslog server is enabled
forward	boolean	no	„1” or “true”, when forwarding of loggings from OMM running on a PC to syslog server is enabled
ipAddr	string	no	IP address of syslog server
port	integer	no	Port number of syslog server

The reply to this request is an object called *SetSyslogServerResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.9.3 EventSyslogServerCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, when syslog server is enabled
forward	boolean	no	„1” or “true”, when forwarding of loggings from OMM running on a PC to syslog server is enabled
ipAddr	string	no	IP address of syslog server, if changed
port	integer	no	Port number of syslog server, if changed

Available filters: (none)

4.10 Flash Memory Management

4.10.1 GetFlashMemUsage

With this request the client can query the fill level of the configuration flash memory.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetFlashMemUsageResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
level	integer	yes	Fill level in percent, -1 if the fill level cannot be calculated

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.11 System Voicebox Number Configuration And Maintenance

4.11.1 GetSysVoiceboxNum

This request is sent from the client to the OMM.

With this request the client can query the current setting of the system voice box number.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

If the change was successful, the reply contains all attributes which *GetSysVoiceboxNum* would return.

The reply to this request is an object called *GetSysVoiceboxNumResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
voiceboxNum	string	yes	System wide voice box number.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.11.2 SetSysVoiceboxNum

This request is sent from the client to the OMM.

With this request the client can set the system voice box number configuration.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

voiceboxNum	string	yes	System wide voice box number.
-------------	--------	-----	-------------------------------

The reply to this request is an object called *SetSysVoiceboxNumResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.11.3 EventSysVoiceboxNumCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
voiceboxNum	string	yes	System wide voice box number.

Available filters: (none)

4.12 System ToneScheme Configuration And Maintenance

4.12.1 ToneSchemeType

This type contains all values of system tone scheme. It is used in different event and responses defined in this chapter.

These are possible values for *ToneSchemeType* type:

Value	Description
AU	Australian tone scheme
AT	Austrian tone scheme, default value (DE) is used
BE	Belgian tone scheme
BY	Belorussian tone scheme, default value (DE) is used
BR	Brazilian tone scheme
CH	Swiss tone scheme
CZ	Czech tone scheme
DK	Danish tone scheme
ES	Spanish tone scheme

EE	Estonian tone scheme, default value (DE) is used
FI	Finnish tone scheme
FR	French.tone scheme
GB	UK-based tone scheme
DE	German tone scheme, this is the default value
HU	Hungarian tone scheme, default value (DE) is used
IT	Italian tone scheme
LT	Lithuanian tone scheme, default value (DE) is used
LV	Latvian tone scheme, default value (DE) is used
NL	Dutch tone scheme
NO	Norwegian tone scheme, default value (DE) is used
PL	Polish tone scheme
RU	Russian tone scheme
SK	Slovakian tone scheme, default value (DE) is used
SE	Swedish tone scheme
TW	Taiwanese tone scheme
UA	Ukrainian tone scheme, default value (DE) is used
US	American tone scheme

4.12.2 GetSysToneScheme

This request is sent from the client to the OMM.

With this request the client can query the current setting of the system tone scheme.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

If the change was successful, the reply contains all attributes which *GetSysToneScheme* would return.

The reply to this request is an object called *GetSysToneSchemeResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

toneScheme	ToneSchemeType	yes	System wide tone scheme.
------------	----------------	-----	--------------------------

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.12.3 SetSysToneScheme

This request is sent from the client to the OMM.

With this request the client can set the system tone scheme configuration.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
toneScheme	ToneSchemeType	yes	System wide tone scheme.

The reply to this request is an object called *SetSysToneSchemeResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.12.4 EventSysToneSchemeCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
toneScheme	ToneSchemeType	yes	System wide tone scheme.

Available filters: (none)

4.13 Automatic Database Backup And Restore Management

4.13.1 GetAutoDBBackup

This request is sent from the client to the OMM.

With this request the client can query common information about the automatic database backup configuration.

Following fields are defined for the request *GetAutoDBBackup* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetAutoDBBackup* is an object called *GetAutoDBBackupResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Server configuration settings where the database file to be put.

The URL contains the path (directory) where the database files (“<yymmdd>_omm_conf.gz”) automatically be put to. The current state of the automatic database backup can be retrieved with the *GetDbTransferState* request.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.13.2 SetAutoDBBackup

This request is sent from the client to the OMM.

With this request the client can set common information about the automatic database backup configuration.

Following fields are defined for the request *SetAutoDBBackup* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Server configuration settings where the database file to be put.

The URL contains the path (directory) where the database files (“<yymmdd>_omm_conf.gz”) automatically be put to.

The reply to *SetAutoDBBackup* is an object called *SetAutoDBBackupResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.13.3 GetAutoDBBackupFileName

With this request a client can get the file name which is used by the OMM to create an automatic database backup.

This file name is created at runtime. It contains, among others, the current date. So be aware that this name will change over time.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to this request is an object called *GetAutoDBBackupFileNameResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
fileName	string	yes	File name

The client needs one of these permissions to use this request: *AllCnfRead*

4.13.4 EventAutoDBBackupCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *AutoDBCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	yes	Changed attributes.

Available filters: (none)

4.13.5 EventAutoDBBackupFileNameCnf

This event is sent by OM AXI when the file name for an automatic database backup has changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *AutoDBCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
fileName	string	no	File name

Available filters: (none)

4.14 Manual Database Backup And Restore Management

4.14.1 ManualDBBackup

This request is sent from the client to the OMM.

With this request the client can trigger a manual database backup action. For database backup to a local file system see 4.74.1.

Following fields are defined for the request *ManualDBBackup* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Server configuration settings where the database file shall be stored.

The URL contains the path and filename where the database files shall be stored.

The reply to *ManualDBBackup* is an object called *ManualDBBackupResp*. It confirms the action indication. The current state of this started action can be retrieved with the *GetDbTransferState* request.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.14.2 ManualDBRestore

This request is sent from the client to the OMM.

With this request the client can trigger a manual database restore action. For database restore from a local file system see 4.74.2.

Following fields are defined for the request *ManualDBRestore* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	URL where the database file shall be retrieved.

The reply to *ManualDBRestore* is an object called *ManualDBRestoreResp*. It confirms the action indication. The current state of this started action can be retrieved with the *GetDbTransferState* request.

Note: A database restore leads to an OMM reset when the action succeeds.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.14.3 DBBackupToUSB

This request is sent from the client to the OMM.

With this request the client can trigger a database backup to RFP USB memory device. Following fields are defined for the request *DBBackupToUSB* in addition to the common attributes:

Name	Type	Mandatory	Description

The reply to *DBBackupToUSB* is an object called *DBBackupToUSBResp*. It confirms the action indication.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*. See dependencies in 5 for this request.

4.14.4 DBRestoreFromUSB

This request is sent from the client to the OMM.

With this request the client can trigger a database restore from RFP USB memory device. Following fields are defined for the request *DBRestoreFromUSB* in addition to the common attributes:

Name	Type	Mandatory	Description

The reply to *DBRestoreFromUSB* is an object called *DBRestoreFromUSBResp*. It confirms the action indication.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*. See dependencies in 5 for this request.

4.15 Database Backup And Restore State Management

4.15.1 GetDbTransferState

This request is sent from the client to the OMM.

With this request the client can retrieve information about the manual/automatic database backup/restore action.

Following fields are defined for the request *GetDbTransferState* in addition to the common attributes:

Name	Type	Mandatory	Description
select	SelectType	yes	Specifies the kind of information to get

SelectType contains types to select information about a database backup/restore action.

These are possible values for the *SelectType* type:

Value	Description
manualBackup	Selects the information for the manual backup operation
manualRestore	Selects the information for the manual restore operation
automaticBackup	Selects the information for the automatic backup operation

The reply to *GetDbTransferState* is an object called *GetDbTransferStateResp*. It contains following fields in addition to the common attributes:

Name	Type	Description
select	SelectType	Specifies the kind of got information
state	string	Status, one of "idle", "running", "finished", "identical", "internal_error", "invalid_file", "invalid_param" or "unknown"
error	string	Possible error string of the last operation
lastupd	string	Time stamp of the last operation

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.15.2 EventDbTransferState

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *AutoDBCnf* or *SystemState*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Description
select	SelectType	Specifies the kind of got information
state	string	Status, one of "idle", "running", "finished", "identical", "internal_error", "invalid_file", "invalid_param" or "unknown"
error	string	Possible error string of the last operation
lastupd	string	Time stamp of the last operation

url	string	Database URL to be transferred
-----	--------	--------------------------------

Available filters: (none)

4.16 Time Zone Configuration And Maintenance

With these requests the time zone can be configured.

4.16.1 TimeZoneListType

This type contains all data of a time zone list element.

Name	Type	Description
id	string	Time zone ID, format like "CET"
name	string	Human readable zone name, in English
modified	boolean	„1” or “true”, if this time zone name has been changed against the time zone name default setting

4.16.2 TimeZoneType

This type contains all data of a time zone, including DST (daylight saving time).

Name	Type	Description
id	string	Time zone ID, format like "CET"
name	string	Human readable zone name, in English
stdOffset	integer	Offset of standard time to UTC in minutes
stdMonth	integer	Start month of standard time, 0..12, 0 means not used
stdDay	integer	Start day of standard time, 0..31, 0 means not used
stdDoW	integer	Start day of week of standard time, 0..7, 0 means not used
stdWoM	integer	Start week of month of standard time, 0..5, 0 means not used, 1 means first week, 5 means last week
stdHour	integer	Start hour of standard time, 0..23
stdMin	integer	Start minute of standard time, 0..59
dstOffset	integer	Offset of DST to UTC in minutes
dstMonth	integer	Start month of DST, 0..12, 0 means not used
dstDay	integer	Start day of DST, 0..31, 0 means not used
dstDoW	integer	Start day of week of DST, 0..7, 0 means not used

dstWoM	integer	Start week of month of DST, 0..5, 0 means not used, 1 means first week, 5 means last week
dstHour	integer	Start hour of DST, 0..23
dstMin	integer	Start minute of DST, 0..59

Note that only id is a mandatory parameter in *SetTimeZoneDetails*.

4.16.3 GetTimeZone

With this request the client can find out the current time zone setting.

Following fields are defined for the request *GetTimeZone* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetTimeZoneResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
id	string	yes	Time zone ID, format like "CET"

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.16.4 GetTimeZoneList

With this request the client can find out which time zone IDs are supported.

Following fields are defined for the request *GetTimeZone* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetTimeZoneListResp*. It contains following elements in addition to the common attributes:

Name	Type	Mandatory	Description
zone	sequence of TimeZoneListType	yes	List of time zone Ids, names and modify flags.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.16.5 EventTimeZoneList

This event is sent by OM AXI when a time zone configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
zone	TimeZoneListType	yes	Time zone ID, name and modify flags that has been changed.

Available filters: (none)

4.16.6 GetTimeZoneDetails

With this request the client can acquire the details of a time zone.

Following fields are defined for the request *GetTimeZoneDetails* in addition to the common attributes:

Name	Type	Mandatory	Description
id	string	yes	Time zone ID, format like "CET"

The reply is an object called *GetTimeZoneDetailsResp*. It contains following elements in addition to the common attributes:

Name	Type	Mandatory	Description
zone	TimeZoneType	yes	Complete data set of this time zone.

Possible values for *errCode*:

Value	Description
ENoEnt	Unknown time zone ID

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.16.7 SetTimeZone

With this request the client can set the active time zone.

Following fields are defined for the request *SetTimeZone* in addition to the common attributes:

Name	Type	Mandatory	Description
id	string	yes	Time zone ID, format like "CET"

The reply is an object called *SetTimeZoneResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

Value	Description
ENoEnt	Unknown time zone ID

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.16.8 SetTimeZoneDetails

With this request the client can modify the details of a time zone.

Following element is defined for the request *SetTimeZoneDetails* in addition to the common attributes:

Name	Type	Mandatory	Description
zone	TimeZoneType	yes	Modified time zone.

The reply is an object called *SetTimeZoneResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
zone	TimeZoneType	no	In case of success: values actually set

Possible values for *errCode*:

Value	Description
ENoEnt	Unknown time zone ID

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.16.9 ResetTimeZoneDetails

With this request the client can reset the time zone details to their default values.

Following fields are defined for the request *ResetTimeZoneDetails* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *ResetTimeZoneDetailsResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.16.10 EventTimeZoneCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
id	string	yes	Changed time zone configuration

Available filters: (none)

4.16.11 EventTimeZoneDetailsCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
zone	TimeZoneType	yes	Changed time zone details configuration

Available filters: (none)

4.17 SNMP Configuration And Maintenance

4.17.1 GetSNMP

With this request the client can query the configuration of SNMP.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Description
		(none)

The reply is an object called *GetSNMPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
readCommunity	string	yes	Read-only community
contact	string	yes	System contact
enableTraps	boolean	yes	„1” or “true”, if trap handling is enabled
trapCommunity	string	yes	Trap community
trapHostAddr	string	yes	IP address of trap host

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.17.2 SetSNMP

With this request the client can configure the SNMP settings.

Only attributes which have to be changed need to be specified.

Following fields are defined for the request *SetSNMP* in addition to the common attributes:

Name	Type	Mandatory	Description
readCommunity	string	no	Read-only community
contact	string	no	System contact
enableTraps	boolean	no	„1” or “true”, if trap handling is enabled
trapCommunity	string	no	Trap community
trapHostAddr	string	no	IP address of trap host

If the change was successful, the reply contains all attributes which *GetSNMP* would return.

The reply is an object called *SetSNMPResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.17.3 EventSNMPCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
readCommunity	string	no	Read-only community
contact	string	no	System contact
enableTraps	boolean	no	„1” or “true”, if trap handling is enabled
trapCommunity	string	no	Trap community
trapHostAddr	string	no	IP address of trap host

Available filters: (none)

4.18 Remote System Dump Configuration And Maintenance

4.18.1 ActivateRemoteSystemDump

This request is sent from the client to the OMM.

With this request the client can activate a remote system dump to the configured destination.

Following fields are defined for the request *ActivateRemoteSystemDump* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetRemoteSystemDump* is an object called *ActivateemoteSystemDumpResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The client needs one of these permissions to use this request: *AllCnfRead*

4.18.2 GetRemoteSystemDump

This request is sent from the client to the OMM.

With this request the client can query for the settings of the remote system dump to the configured destination.

Following fields are defined for the request *GetRemoteSystemDump* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetRemoteSystemDump* is an object called *GetRemoteSystemDumpResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, system dump shall be uploaded to a remote server daily on time.
hour	integer	yes	Hour time value to upload the remote system dump.
minute	integer	yes	Minute time value to upload the remote system dump.
url	URLType	yes	Server configuration settings where the remote system dump to be put.

The URL contains the path (directory) where the database file (“sys_dump.txt.gz”) be put to.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.18.3 SetRemoteSystemDump

This request is sent from the client to the OMM.

With this request the client can set the settings of the remote system dump to the configured destination.

Following fields are defined for the request *SetRemoteSystemDump* in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, system dump shall be uploaded to a remote server daily on time.

hour	integer	no	Hour time value to upload the remote system dump.
minute	integer	no	Minute time value to upload the remote system dump.
url	URLType	no	Server configuration settings where the remote system dump to be put.

The URL contains the path (directory) where the database file (“sys_dump.txt.gz”) be put to.

The reply to *SetRemoteSystemDump* is an object called *SetRemoteSystemDumpResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.18.4 EventRemoteSystemDumpCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, system dump shall be uploaded to a remote server daily on time.
hour	integer	no	Hour time value to upload the remote system dump.
minute	integer	no	Minute time value to upload the remote system dump.
url	URLType	no	Server configuration settings where the remote system dump to be put.

Available filters: (none)

4.19 Health State Monitoring

4.19.1 GetHealthState

With this request the client can query the health state of the OpenMobility® system.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetHealthStateResp*. It contains a sequence of elements called *health* in addition to the common attributes:

Name	Type	Mandatory	Description
health	sequence of HealthType	yes	Health data of a component

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.19.2 EventHealthState

This event is sent by OM AXI when the health state of the OpenMobility® system has changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf* or *SystemState*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification are the same as in *GetHealthState*(4.19.1), but only the attributes which have changed are added and only one *HealthType* element is contained.

Available filters: (none)

4.20 Event Log Monitoring

4.20.1 EventLogEntryType

This type contains all data of a log entry.

Name	Type	Description
id	integer	Event log entry id
time	string	Timestamp of this/last entry
level	enumeration	One of: <i>Always, GenInfo, FnEnter, Event, MinorErr, MajorErr, FatalErr</i>
name	string	Name of subsystem which wrote this message
count	integer	How often this message occurred in a row
msg	string	Log text

4.20.2 GetEventLogBuffer

With this request the client can get the current content of the event log buffer. As the buffer may be quite large, it must be fetched separated into several chunks.

Following fields are defined for this Request in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	Event log entry id to start from
maxRecords	integer, default = 20	no	Maximal number of records to return. Not more than 20 allowed. 0 is not allowed here.

When the client sends this request the first time, it must set *id* to 0. The response will contain 0 to 10 entries and a new *id* value to be used to get the next chunk of the buffer. The last response has *eof* set.

The reply is an object called *GetEventLogBufferResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	no	Event log entry index to be used next time
eof	boolean	no	„1” or “true”, if there are no additional entries
e	sequence of EventLogEntryType	no	0 to 20 entries

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.20.3 DeleteEventLogBuffer

With this request the client can delete the current OMM event log buffer.

Following fields are defined for the request *DeleteEventLogBuffer* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *DeleteEventLogBufferResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.20.4 EventEventLogEntry

This event is sent by OM AXI when a new log entry has been done.

To get this event the client has to subscribe to *EventLog*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
e	EventLogEntryType	yes	New log entry

Available filters: (none)

4.21 External User Data Server Configuration And Maintenance

4.21.1 GetUserDataServer

This request is sent from the client to the OMM.

With this request the client can query common information about the user data server configuration.

Following fields are defined for the request *GetUserDataServer* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetUserDataServer* is an object called *GetUserDataServerResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	User data server configuration settings
useCommonFileNameOnServer	boolean	yes	„1” or “true”, if the common file name “ <i>user.cfg</i> ” is used to be got from the server instead of “ <i><number>.cfg</i> ” or “ <i><loginId>.cfg</i> ”. This value is applied when the user credential access to the server is used. This value is read only. It can only be set within the <i>user_common.cfg</i> file got from the server.

The URL contains the path to the user data files.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.21.2 SetUserDataServer

This request is sent from the client to the OMM.

With this request the client can set the configuration of the User Data Server.

Following fields are defined for the request *SetUserDataServer* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	no	User data server configuration settings.

The reply to *SetUserDataServer* is an object called *SetUserDataServerResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.21.3 EventUserDataServerCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	no	Changed attributes.
useCommonFileNameOnServer	boolean	no	„1” or “true”, if the common file name “user.cfg” is used to be got from the server instead of “<number>.cfg” or “<loginId>.cfg”. This value is applied when the user credential access to the server is used. This value is read only. It can only be set within the <i>user_common.cfg</i> file got from the server.

Available filters: (none)

4.21.4 GetPreserveUserDeviceRelation

This request is sent from the client to the OMM.

With this request the client can query the user device relation behavior configuration for OMM start up when a new database is imported.

Following fields are defined for the request *GetPreserveUserDeviceRelation* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetPreserveUserDeviceRelation* is an object called *GetPreserveUserDeviceRelationResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, if the user device relation between users and devices are preserved at OMM start up when a new database is imported.

The URL contains the path to the user data files.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.21.5 SetPreserveUserDeviceRelation

This request is sent from the client to the OMM.

With this request the client can set the user device relation behavior for OMM start up when a new database is imported.

Following fields are defined for the request *SetPreserveUserDeviceRelation* in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

enable	boolean	no	„1” or “true”, if the user device relation between users and devices are preserved at OMM start up when a new database is imported.
--------	---------	----	---

The reply to *SetPreserveUserDeviceRelation* is an object called *SetPreserveUserDeviceRelationResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.21.6 EventPreserveUserDeviceRelationCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, if the user device relation between users and devices are preserved at OMM start up when a new database is imported.

Available filters: (none)

4.22 System Configuration Files And Maintenance For Provisioning

4.22.1 GetSystemProvUpdTrig

With this request the client can query the system provisioning update trigger settings to update configuration files. These can be “ipdect.cfg”, “mac.cfg”, “ima.cfg”, “customer_image.png” and the Mitel SIP-DECT software image. For more information see Mitel SIP-DECT system manual.

Following fields are defined for the request *GetSystemProvUpdTrig* in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, configuration files shall be updated from a server daily on time.
hour	integer	yes	Hour time value to update configuration files.
minute	integer	yes	Minute time value to update configuration files.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.22.2 SetSystemProvUpdTrig

With this request the client can set the system provisioning update trigger settings to update provider configuration files. These can be “ipdect.cfg”, “mac.cfg”, “ima.cfg”, “customer_image.png” and the Mitel SIP-DECT software image. For more information see Mitel SIP-DECT system manual.

Following fields are defined for the request *SetSystemProvUpdTrig* in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, configuration files shall be updated from a server daily on time.
hour	integer	no	Hour time value to update configuration files.
minute	integer	no	Minute time value to update configuration files.

The reply is an object called *SetSystemProvUpdTrigResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.22.3 EventSystemProvUpdTrigCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, configuration files shall be updated from a server daily on time.
hour	integer	no	Hour time value to update configuration files.
minute	integer	no	Minute time value to update configuration files.

Available filters: (none)

4.22.4 GetSystemCredentials

With this request the client can query the system credential settings for provisioning.

Following fields are defined for the request *GetSystemCredentials* in addition to the common attributes:

Name	Type	Mandatory	Description
username	string	yes	System credential user name to access provider configuration files.
password	string	yes	System credential password to access provider configuration files, encrypted with public key.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.22.5 SetSystemCredentials

With this request the client can set the system credential settings for provisioning.

Following fields are defined for the request *SetSystemCredentials* in addition to the common attributes:

Name	Type	Mandatory	Description
username	string	no	System credential user name to access provider configuration files.
password	string	no	System credential password to access provider configuration files, encrypted with public key.

The reply is an object called *SetSystemCredentialsResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.22.6 EventSystemCredentialsCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
username	string	no	System credential user name to access provider configuration files.
password	string	no	System credential password to access provider configuration files, encrypted with public key.

Available filters: (none)

4.22.7 GetConfigURL

With this request the client can query the system configuration URL settings used to load configuration files. These can be "ipdect.cfg", "mac.cfg", "ima.cfg", "customer_image.png" and the Mitel SIP-DECT software image. For more information see Mitel SIP-DECT system manual.

Following fields are defined for the request *GetConfigURL* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to load configuration files from the external server.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.22.8 SetConfigURL

With this request the client can set the URL settings used to load configuration files.

Following fields are defined for the request *SetConfigURL* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to load configuration files from the external server.

The reply is an object called *SetConfigURLResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.22.9 EventConfigURLCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to load configuration files from the external server.

Available filters: (none)

4.22.10 GetUsedConfigURL

With this request the client can query the used system configuration URL settings to load configuration files. These can be "ipdect.cfg", "mac.cfg", "ima.cfg", "customer_image.png" and the Mitel SIP-DECT software image. For more information see Mitel SIP-DECT system manual.

Following fields are defined for the request *GetUsedConfigURL* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Used configuration URL settings to load the configuration files from the external server.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.22.11 EventUsedConfigURL

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	yes	Used configuration URL settings to load the configuration files from the external server.

Available filters: (none)

4.23 User Data Import And Maintenance

4.23.1 RequestManualUserDataImport

This request is sent from the client to the OMM.

With this request the client can retrieve trigger an import from the user data server.

Following fields are defined for the request *RequestManualUserDataImport* in addition to the common attributes:

Name	Type	Mandatory	Description
userId	string	no	Specifies the user data file (" <i><userId>.cfg</i> ") to get from the server. If the <i>userId</i> is not used all user data sets known by the OMM will be got and even the "user_common.cfg" file

The reply to *RequestManualUserDataImport* is an object called *RequestManualUserDataImportResp*. It contains following fields in addition to the common attributes:

Name	Type	Description

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.23.2 EventUserDataImport

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *UserDataImport*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Description
filename	string	Specifies the filename to be imported, one of " <i><userId>.cfg</i> " or "user_common.cfg".
result	string	Result of the user data import, one of "Imported" or "Failed"

Available filters: (none)

4.24 Network Parameters Configuration And Maintenance

4.24.1 NetParamType

This type contains all data fields of the network parameters. It is used in different requests and responses defined in this chapter. Not all fields are used in all OMM versions and in all Request and Response types.

Name	Type	Mandatory	Description
voiceToS	integer	yes	ToS for voice packets, 0..255
sigToS	integer	yes	ToS for signalling packets, 0..255
ttl	integer	yes	Time To Life
voiceEthPrio	integer	no	802.1p priority for voice packets, 0..7
sigEthPrio	integer	no	802.1p priority for signalling packets, 0..7

The attributes *voiceEthPrio* and *sigEthPrio* are mandatory on OMM versions which support this feature. In the *EventNetParamsCnf* notification all elements are optional.

4.24.2 GetNetParams

With this request the client can get some properties of the network interface.

Following fields are defined for the request *GetNetParams* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetNetParamsResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
net	NetParamType	yes	Network parameters

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.24.3 SetNetParams

With this request the client can set some properties of the network interface.

Following element is defined for the request *SetNetParams* in addition to the common attributes:

Name	Type	Mandatory	Description
net	NetParamType	yes	Network parameters

The reply is an object called *SetNetParamsResp*. It contains all actual set attributes, all attributes are optional.

These attributes are filled in only if the request succeeded. They contain the values actually set in the OMM.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.24.4 EventNetParamsCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
net	NetParamType	yes	Contains changed attributes

Available filters: (none)

4.25 PARK / SARI Configuration And Maintenance

4.25.1 GetPARK

With this request the client can query the PARK.

Following fields are defined for the request *GetPARK* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetPARKResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
park	string	yes	PARK in following form: "1F10187326"
initialPARK	string	yes	Initial PARK, alternative representation

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.25.2 SetPARK

With this request the client can set the PARK.

Consider the license restrictions for this command (see 5.5).

Following fields are defined for the request *SetPARK* in addition to the common attributes:

Name	Type	Mandatory	Description
park	string	yes	PARK in following form: "1F10187326"

When this request succeeds, the OMM may reboot automatically.

The reply is an object called *SetPARKResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

Value	Description
EForbidden	PARK is part of the license and cannot be set

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.25.3 EventPARKCnf

This is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.2 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
park	string	yes	Changed PARK configuration
initialPARK	string	yes	Changed initial PARK, alternative representation

Available filters: (none)

4.25.4 PARKFromServer

With this request the client can initiate a PARK online request to get a new PARK form the PARK server. Following fields are defined for the request *PARKFromServer* in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

When this request succeeds, the OMM may reboot automatically.

The reply is an object called *PARKFromServerResp*.

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.25.5 EventPARKFromServerResult

This event is sent by OM AXI when the result from PARK server is available. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *ParkServerResult* This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
result	string	yes	Result from PARK server online request. Following results are possible:

			"Success" // Operation successfully performed "RequestAlreadyStarted" // File transfer is already running "NoOwnMACAddress" // Cannot retrieve on system MAC address "FileTransferError" // Any FTS error "ServerResponseError" // Server response with error code instead of a PARK "InvalidPARK" // Received PARK is not valid "WrongMAC" // MAC address in PARK.xml file does not fit to the system "WrongChecksum" // Modified PARK.xml file "MissingParam" // Missing PARK, MAC or checksum in PARK.xml "FileError" // Error opening or reading the file
--	--	--	--

Available filters: (none)

4.25.6 GetSARI

With this request the client can query the SARI.

Following fields are defined for the request *GetSARI* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetPARKResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
sari	string	yes	SARI in following form: "1F10187326"

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.25.7 SetSARI

With this request the client can set the SARI.

Consider the license restrictions for this command (see 5.5).

Following fields are defined for the request *SetSARI* in addition to the common attributes:

Name	Type	Mandatory	Description
sari	string	yes	SARI in following form: "1F10187326"

When this request succeeds, the OMM may reboot automatically.

The reply is an object called *SetSARIResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.25.8 EventSARICnf

This is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.2 on how to set up a notification session.

To get this event the client has to subscribe to *Roaming*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
sari	string	yes	Changed SARI configuration

Available filters: (none)

4.26 DECT Encryption Configuration And Maintenance

4.26.1 GetDECTEncryption

With this request the client can query the state of DECT encryption.

Following fields are defined for the request *GetDECTEncryption* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetDECTEncryptionResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, if encryption is enabled

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.26.2 SetDECTEncryption

With this request the client can enable or disable DECT encryption.

Following fields are defined for the request *SetDECTEncryption* in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, to enable encryption

The reply is an object called *SetDECTEncryptionResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.26.3 EventDECTEncryptionCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	yes	Changed DECT encryption configuration

Available filters: (none)

4.27 DECT Regulatory Domain Configuration And Maintenance

4.27.1 DECTRegDomainType

This type contains values for different DECT regulatory domains.

These are possible values:

Value	Description
None	Not configured
EMEA	EMEA
US	US (FCC/IC)
Brazil	Brazil (Anatel)
Taiwan	Taiwan
Radio1910_1927MHz250mW	Radio 1910-1927MHz with 250mW

4.27.2 GetDECTRegDomain

With this request the client can query the active DECT regulatory domain.

Following fields are defined for the request *GetDECTRegDomain* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetDECTRegDomainResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
regDomain	DECTRegDomainType	yes	Active DECT regulatory domain

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.27.3 SetDECTRegDomain

With this request the client can set the active DECT regulatory domain.

Following fields are defined for the request *SetDECTRegDomain* in addition to the common attributes:

Name	Type	Mandatory	Description
regDomain	DECTRegDomainType	yes	Active DECT regulatory domain

The reply is an object called *SetDECTRegDomainResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.27.4 EventDECTRegDomainCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
regDomain	DECTRegDomainType	yes	Changed DECT regulatory domain

Available filters: (none)

4.28 DECT Paging Area Size Configuration And Maintenance

4.28.1 GetDECTPagingAreaSize

With this request the client can query the DECT paging area size.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetDECTAuthCodeResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
size	integer	yes	Paging area size

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.28.2 SetDECTPagingAreaSize

With this request the client can set the DECT paging area size.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

size	integer	yes	Paging area size, one of 16, 32, 64, 128, 256
------	---------	-----	---

The reply is an object called *GetDECTPagingAreaSizeResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.28.3 EventDECTPagingAreaCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
size	integer	no	Changed DECT paging area size

Available filters: (none)

4.29 DECT Authentication Code Configuration And Maintenance

4.29.1 GetDECTAuthCode

With this request the client can query the global DECT authentication code.

Following fields are defined for the request *GetDECTAuthCode* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetDECTAuthCodeResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
ac	string	yes	DECT authentication code

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.29.2 SetDECTAuthCode

With this request the client can set the global DECT authentication code.

Following fields are defined for the request *SetDECTAuthCode* in addition to the common attributes:

Name	Type	Mandatory	Description
ac	string	yes	DECT authentication code

The reply is an object called *SetDECTAuthCodeResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.29.3 EventDECTAuthCodeCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
ac	string	yes	Changed DECT authentication code

Available filters: (none)

4.30 DECT Subscription Mode Configuration And Maintenance

4.30.1 GetDECTSubscriptionMode

With this request the client can query the current state of the DECT subscription mode.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to this request is an object called *GetDECTSubscriptionModeResp*. It contains following fields in addition to the common attributes:

Name	Type	Description
mode	DECTSubscriptionModeType	Same modes as defined in chapter 4.30.2.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.30.2 SetDECTSubscriptionMode

A client can send this request to the OMM to enable or disable DECT subscription modes. Once started, the subscription modes will be stopped by timers in the OMM. Additionally, they can be stopped explicitly with another *SetDECTSubscriptionMode* request.

Use the mode *Wildcard* to enable wildcard subscriptions, i.e. subscriptions of DECT phones which's IPEIs are not configured in the OMM yet. Use *Configured* to enable subscription of configured IPEIs only. Disable any subscription by using mode *Off*.

Enabling *Wildcard* will also implicitly allow the *Configured* mode.

The *Configured* mode does always have a timeout of 24 hours, no matter if it was started explicitly or implicitly with *Wildcard*. These 24 hours will not be extended automatically.

The *Wildcard* mode can be started with a timeout between 1 minute and 60 minutes. Each time a wildcard subscription actually took place, the timeout will be extended by the time interval once given in the request.

Activating one of these Subscription modes while there still a subscription mode enabled will have the same result as stopping the old mode before starting the new mode.

To keep track of the states of subscription modes, there is a notification event defined, refer to chapter 4.30.3.

Following fields are defined for the request *SetDECTSubscriptionMode* in addition to the common attributes:

Name	Type	Mandatory	Description
mode	DECTSubscriptionModeType	yes	Mode to be set.
timeout	integer	no	Timeout for <i>Wildcard</i> mode. Default 3 minutes. Ignored in other modes.

Possible values for *mode* (DECTSubscriptionModeType):

Value	Description
Wildcard	Start Wildcard subscription mode.
Configured	Start Configured subscription mode.
Off	Stop all subscription modes.

The reply to this request is called *SetDECTSubscriptionModeResp*. It contains all actual set attributes, all attributes are optional.

Note: Setting the DECT subscription mode is an asynchronous action and can take a little time. Therefore the response does not contain the set value. The event *EventDECTSubscriptionMode* notifies the client when the value was set.

Possible values for *errCode*:

Value	Description
EInval	Unknown subscription mode or timeout out of range.

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.30.3 EventDECTSubscriptionMode

This event is sent by OM AXI when a subscription mode is started or stopped in the OMM. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *DECTSubscriptionMode*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
mode	DECTSubscriptionModeType	yes	Same modes as defined in chapter 4.39.8.

Available filters: (none)

4.31 DECT Phone Settings Configuration And Maintenance

4.31.1 PpBootTestType

The *PpBootTestType* type contains the following entries.

Name	Type	Description
enable	boolean	Enables or disables the DECT phone text.
text	string	Text to be used at the DECT phone.

4.31.2 GetDECTPpSettings

With this request the client can query the system wide DECT phone settings.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetDECTPpSettingsResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
bootTextHeadline	PpBootTestType	yes	Boot headline text configuration to be used for all DECT phones.
bootTextStartup	PpBootTestType	yes	Boot startup text configuration to be used for all DECT phones.
dialByNumberOnly	boolean	yes	Selection for the dial editor setting in the DECT phones. „1” or “true”, if dial by number only is enabled at the DECT phones, otherwise the alphanumeric dial editor is used.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.31.3 SetDECTPpSettings

With this request the client can set system wide DECT phone settings.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
bootTextHeadline	PpBootTestType	no	Boot headline text configuration to be used for all DECT phones.
bootTextStartup	PpBootTestType	no	Boot startup text configuration to be used for all DECT phones.
dialByNumberOnly	boolean	no	Selection for the dial editor setting in the DECT phones. „1” or “true”, if dial by number only is enabled at the DECT

			phones, otherwise the alphanumeric dial editor is used.
--	--	--	---

The reply is an object called *SetDECTPpSettingsResp*. It contains all actual set attributes, all attributes are optional (see *GetDECTPpSettingsResp*).

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.31.4 EventDECTPpSettingsCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
bootTextHeadline	PpBootTestType	no	Boot headline text configuration to be used for all DECT phones.
bootTextStartup	PpBootTestType	no	Boot startup text configuration to be used for all DECT phones.
dialByNumberOnly	boolean	no	Selection for the dial editor setting in the DECT phones. „1” or “true”, if dial by number only is enabled at the DECT phones, otherwise the alphanumeric dial editor is used.

Available filters: (none)

4.32 DECT Power Limit Configuration And Maintenance

4.32.1 GetDECTPowerLimit

With this request the client can query the configuration of the DECT power limitation to 100mW.

Following fields are defined for the request *GetDECTPowerLimit* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetDECTPowerLimitResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, if the DECT power limitation to 100 mW is enabled

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.32.2 SetDECTPowerLimit

With this request the client can set the configuration of the DECT power limitation to 100mW.

Following fields are defined for the request *SetDECTPowerLimit* in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, to enable the DECT power limitation to 100 mW

The reply is an object called *SetDECTPowerLimitResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.32.3 EventDECTPowerLimitCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	yes	Changed DECT power limitation to 100 mW configuration

Available filters: (none)

4.33 Auto-Create On Subscription Configuration And Maintenance

4.33.1 GetDevAutoCreate

With this request the client can query if auto-create devices on DECT subscription is enabled.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetDevAutoCreateResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, if auto-create is enabled

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.33.2 SetDevAutoCreate

With this request the client can enable or disable auto-create on subscription.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, to enable auto-create

The reply is an object called *SetDevAutoCreateResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.33.3 EventDevAutoCreateCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	yes	Changed auto-create configuration

Available filters: (none)

4.34 SIP And RTP Settings Configuration And Maintenance

4.34.1 SIP CodecType

CodecType is an enumeration value defined as follows:

Value	Description
None	No codec
G.711-u-law	G.711 μ -law
G.711-A-law	G.711 A-law
G.729-A	G.729 A
G.722	G.722 Hi-Q audio technology (wide band)

4.34.2 SIPTransportProtocolType

SecureSIPTransportProtocolType is an enumeration value defined as follows:

Value	Description
-------	-------------

UDP	Default setting
TCP	
UDPandTCP	
TLS	
PersistentTLS	

4.34.3 SIPPortRangeType

Name	Type	Description
startPort	integer	Start SIP port number of the range for call server registering.
endPort	integer	End SIP port number of the range for call server registering.

4.34.4 GetBasicSIP

With this request the client can query the basic SIP settings.

Following fields are defined for the request *GetBasicSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetBasicSIPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
transportProt	SIPTransportProtocolType	yes	Used transport protocol.
proxyServer	string	yes	Proxy server name or IP address
proxyPort	integer	yes	Proxy server port
outboundProxyServer	string	yes	Outbound proxy server name or IP address, if changed
outboundProxyPort	integer	yes	Outbound proxy server port, if changed
regServer	string	yes	Registrar server name or IP address
regPort	integer	yes	Registrar server port
regPeriod	integer	yes	Registration period in seconds
sipRegisterCheckNum	string	yes	Phone number or SIP user id which is used for the PBX SIP register check. (see also PPuserType for more information)
gruu	boolean	yes	Enables or disables the Globally Routable

			User-Agent URI (GRUU) support according to RFC 5627. Default is enabled.
--	--	--	--

For possible `errCode` values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.34.5 SetBasicSIP

With this request the client can set the basic SIP settings.

Following fields are defined for the request *SetBasicSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
<code>transportProt</code>	<code>SIPTransportProtocolType</code>	yes	Used transport protocol.
<code>proxyServer</code>	string	no	Proxy server name or IP address
<code>proxyPort</code>	integer	no	Proxy server port
<code>outboundProxyServer</code>	string	no	Outbound proxy server name or IP address, if changed
<code>outboundProxyPort</code>	integer	no	Outbound proxy server port, if changed
<code>regServer</code>	string	no	Registrar server name or IP address
<code>regPort</code>	integer	no	Registrar server port
<code>regPeriod</code>	integer	no	Registration period in seconds
<code>gruu</code>	boolean	no	Enables or disables the Globally Routable User-Agent URI (GRUU) support according to RFC 5627. Default is enabled.

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetBasicSIPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
<code>transportProt</code>	<code>SIPTransportProtocolType</code>	yes	Used transport protocol.
<code>proxyServer</code>	string	no	Proxy server name or IP address
<code>proxyPort</code>	integer	no	Proxy server port
<code>outboundProxyServer</code>	string	no	Outbound proxy server name or IP address, if changed
<code>outboundProxyPort</code>	integer	no	Outbound proxy server port, if changed
<code>regServer</code>	string	no	Registrar server name or IP address
<code>regPort</code>	integer	no	Registrar server port

regPeriod	integer	no	Registration period in seconds
sipRegisterCheckNum	string	no	Phone number or SIP user id which is used for the PBX SIP register check. (see also PPuserType for more information)
gruu	boolean	no	Enables or disables the Globally Routable User-Agent URI (GRUU) support according to RFC 5627. Default is enabled.

When this request succeeds and changes are detected, the OMM may reboot automatically.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.34.6 EventBasicSIPcNf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
transportProt	SIPTransportProtocolType	yes	Used transport protocol.
proxyServer	string	no	Proxy server name or IP address, if changed
proxyPort	integer	no	Proxy server port, if changed
outboundProxyServer	string	no	Outbound proxy server name or IP address, if changed
outboundProxyPort	integer	no	Outbound proxy server port, if changed
regServer	string	no	Registrar server name or IP address, if changed
regPort	integer	no	Registrar server port, if changed
regPeriod	integer	no	Registration period in seconds, if changed
sipRegisterCheckNum	string	no	Phone number or SIP user id which is used for the PBX SIP register check. (see also PPuserType for more information)
gruu	boolean	no	Enables or disables the Globally Routable User-Agent URI (GRUU) support according to RFC 5627. Default is enabled.

Available filters: (none)

4.34.7 GetAdvancedSIP

With this request the client can query the advanced SIP settings.

Following fields are defined for the request *GetAdvancedSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetAdvancedSIPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
mwiSubscription	boolean	yes	„1” or “true”, for explicit MWI subscription
userAgentCompatibility	boolean	yes	„1” or “true”, for using the user agent (Aastra-name) compatibility mode. None compatibility mode (false) uses the user agent Mitel name.
userAgentInfo	boolean	yes	„1” or “true”, for sending the UserAgent header field
dialTerminator	string	yes	Two character dial string that separates a dialed number to dialing information and digit treatment processing.
regFailedRetryTimer	integer	yes	Time to retry failed registrations in seconds
regTimeoutRetryTimer	integer	yes	Time to retry registrations in seconds
transactionTimer	integer	yes	Transaction timer in milliseconds
sessionTimer	integer	yes	The time, in seconds, that the OMM uses to send periodic re-INVITE requests to keep a session alive. The proxy uses these re-INVITE request to maintain the status of the connected sessions. Range is 0, 90-86400, default value is 0. See RFC 4028 for details.
blacklistTimeout	integer	yes	Blacklist timeout in minutes
incomingCallTimeout	integer	yes	Incoming call timeout in seconds. Valid values are: 30 to 300, default value is 180.
callerDetermination	enumeration	yes	One of “ <i>P-Assert-Identity</i> ” (default) or “ <i>From/To</i> ”
multipleRing	boolean	yes	„1” or “true”, for multiple 180 ringing support
semiAttendedTransferMode	enumeration	yes	Kind of semi attended transfer. One of “ <i>Blind</i> ” (default) or “ <i>Attended</i> ”
referToWithReplaces	boolean	yes	„1” or “true”, refer in ringing state transfer support, no automatic disconnect. Default is “false”.

			This parameter is only used, when the <i>semiAttendedTransferMode</i> parameter is set to "Blind".
xAastrald	boolean	yes	„1” or “true”, for sending the “X-Aastra-Id” in the SIP protocol
callRejectStateCodeUsr	integer	yes	Specifies the SIP state code which will be send as response when the user rejects an incoming call pressing the “reject”. Valid values are “400” to “699”. The default is “486”
callRejectStateCodeDev	integer	yes	Specifies the SIP state code which will be send as response when the an incoming call have to be rejected because the called DECT DECT phone is unreachable (e.g. the DECT phone is out of range or out of battery). Valid values are “400” to “699”. The default is 486.
contactMatching	enumeration	yes	Specifies the method for which the phone uses to match the contact header in a SIP registration packet. One of “Uri”, “IpOnly”, “IpAndUsername” or “UsernameOnly”.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.34.8 SetAdvancedSIP

With this request the client can set the advanced SIP settings.

Following fields are defined for the request *SetAdvancedSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
mwiSubscription	boolean	no	„1” or “true”, for explicit MWI subscription
userAgentInfo	boolean	no	„1” or “true”, for sending the UserAgent header field
userAgentCompatibility	boolean	no	„1” or “true”, for using the user agent (Aastra-name) compatibility mode (default=false). None compatibility mode (false) uses the user agent Mitel name.
dialTerminator	string	no	Two character dial string that separates a dialed number to dialing information and digit treatment processing.
regFailedRetryTimer	integer	no	Time to retry failed registrations in seconds
regTimeoutRetryTimer	integer	no	Time to retry registrations in seconds
transactionTimer	integer	no	Transaction timer in milliseconds
sessionTimer	integer	no	The time, in seconds, that the OMM uses to send periodic re-INVITE requests to keep a session alive. The proxy uses these re-INVITE request to

			maintain the status of the connected sessions. Range is 0, 90-86400, default value is 0. See RFC 4028 for details.
incomingCallTimeout	integer	no	Incoming call timeout in seconds. Valid values are: 30 to 300, default value is 180.
blacklistTimeout	integer	no	Blacklist timeout in minutes
callerDetermination	enumeration	no	One of "P-Assert-Identity" (default) or "From/To"
multipleRing	boolean	no	„1" or "true", for multiple 180 ringing support
semiAttendedTransferMode	enumeration	no	Kind of semi attended transfer. One of "Blind" (default) or "Attended"
referToWithReplaces	boolean	no	„1" or "true", refer in ringing state transfer support, no automatic disconnect. Default is "false". This parameter is only used, when the <i>semiAttendedTransferMode</i> parameter is set to "Blind".
xAastraId	boolean	no	„1" or "true", for sending the "xAastraId" in the SIP protocol (default=false)
callRejectStateCodeUsr	integer	no	Specifies the SIP state code which will be send as response when the user rejects an incoming call pressing the "reject". Valid values are "400" to "699". The default is "486"
callRejectStateCodeDev	integer	yes	Specifies the SIP state code which will be send as response when the an incoming call have to be rejected because the called DECT DECT phone is unreachable (e.g. the DECT phone is out of range or out of battery). Valid values are "400" to "699". The default is 486.
contactMatching	enumeration	no	Specifies the method for which the phone uses to match the contact header in a SIP registration packet. One of "Uri", "IpOnly", "IpAndUsername" or "UsernameOnly".

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetAdvancedSIPResp*. It contains all actual set attributes, all attributes are optional.

When this request succeeds, the OMM may reboot automatically.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.34.9 EventAdvancedSIPCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
mwiSubscription	boolean	no	„1” or “true”, for explicit MWI subscription, if changed
userAgentInfo	boolean	no	„1” or “true”, for sending the UserAgent header field, if changed
userAgentCompatibility	boolean	yes	„1” or “true”, for using the user agent (Aastra-name) compatibility mode. None compatibility mode (false) uses the user agent Mitel name.
dialTerminator	string	no	Two character dial string that separates a dialed number to dialing information and digit treatment processing.
regFailedRetryTimer	integer	no	Time to retry failed registrations in seconds, if changed
regTimeoutRetryTimer	integer	no	Time to retry registrations in seconds
transactionTimer	integer	no	Transaction timer in milliseconds, if changed
sessionTimer	integer	no	The time, in seconds, that the OMM uses to send periodic re-INVITE requests to keep a session alive. The proxy uses these re-INVITE request to maintain the status of the connected sessions. Range is 0, 90-86400, default value is 0. See RFC 4028 for details.
blacklistTimeout	integer	no	Blacklist timeout in minutes, if changed
incomingCallTimeout	integer	no	Incoming call timeout in seconds. Valid values are: 30 to 300, default value is 180.
callerDetermination	enumeration	no	One of “P-Assert-Identity” (default) or “From/To”
multipleRing	boolean	no	„1” or “true”, for multiple 180 ringing support
semiAttendedTransferMode	enumeration	no	Kind of semi attended transfer. One of “Blind” (default) or “Attended”
referToWithReplaces	boolean	no	„1” or “true”, refer in ringing state transfer support, no automatic disconnect. Default is “false”. This parameter is only used, when the <i>semiAttendedTransferMode</i> parameter is set to “Blind”.
xAastrald	boolean	no	„1” or “true”, for sending the “xAastrald” in the SIP protocol

callRejectStateCodeUsr	integer	no	Specifies the SIP state code which will be send as response when the user rejects an incoming call pressing the “reject”. Valid values are “400” to “699”. The default is “486”
callRejectStateCodeDev	integer	yes	Specifies the SIP state code which will be send as response when the an incoming call have to be rejected because the called DECT DECT phone is unreachable (e.g. the DECT phone is out of range or out of battery). Valid values are “400” to “699”. The default is 486.
contactMatching	enumeration	no	Specifies the method for which the phone uses to match the contact header in a SIP registration packet. One of “Uri”, “IpOnly”, “IpAndUsername” or “UsernameOnly”.

Available filters: (none)

4.34.10 GetSecureSIP

With this request the client can query the secure SIP settings.

Following fields are defined for the request *GetSecureSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetSecureSIPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
keepAliveTimeoutEnable	boolean	yes	„1” or “true”, if keep alive timeout for the persistent TLS protocol is set. Only used for persistent TLS protocol.
timeout	integer	yes	Keep alive timeout for the persistent TLS protocol. Value range 10-3600 seconds, default value is 30 seconds. Only used for persistent TLS protocol.
sendSipsOverTLS	boolean	yes	„1” or “true”, if send secure SIP over TLS is set. Only used for TLS or persistent TLS protocol.
authenticationTLS	boolean	yes	„1” or “true”, if authentication for TLS is set. Only used for TLS or persistent TLS protocol.

commonNameValidationTLS	boolean	yes	„1” or “true”, if common name validation for TLS is set. Only used for TLS or persistent TLS protocol. Ignored when authenticationTLS is not set.
-------------------------	---------	-----	---

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.34.11 SetSecureSIP

With this request the client can set the secure SIP settings.

Following fields are defined for the request *SetSecureSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
keepAliveTimeoutEnable	boolean	no	„1” or “true”, if keep alive timeout for the persistent TLS protocol is set. Only used for persistent TLS protocol.
timeout	integer	no	Keep alive timeout for the persistent TLS protocol. Value range 10-3600 seconds, default value is 30 seconds. Only used for persistent TLS protocol.
sendSipsOverTLS	boolean	no	„1” or “true”, if send secure SIP over TLS is set. Only used for TLS or persistent TLS protocol.
authenticationTLS	boolean	no	„1” or “true”, if authentication for TLS is set. Default value is “1”. Only used for TLS or persistent TLS protocol.
commonNameValidationTLS	boolean	no	„1” or “true”, if common name validation for TLS is set. Default value is “0”. Only used for TLS or persistent TLS protocol. Ignored when authenticationTLS is not set.

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetSecureSIPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

keepAliveTimeoutEnable	boolean	yes	„1” or “true”, if keep alive timeout for the persistent TLS protocol is set. Only used for persistent TLS protocol.
timeout	integer	yes	Keep alive timeout for the persistent TLS protocol. Value range 10-3600 seconds, default value is 30 seconds. Only used for persistent TLS protocol.
sendSipsOverTLS	boolean	yes	„1” or “true”, if send secure SIP over TLS is set. Only used for TLS or persistent TLS protocol.
authenticationTLS	boolean	yes	„1” or “true”, if authentication for TLS is set. Only used for TLS or persistent TLS protocol.
commonNameValidationTLS	boolean	yes	„1” or “true”, if common name validation for TLS is set. Only used for TLS or persistent TLS protocol. Ignored when authenticationTLS is not set.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.34.12 EventSecureSIPCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Description
keepAliveTimeoutEnable	boolean	„1” or “true”, if keep alive timeout for the persistent TLS protocol is set. Only used for persistent TLS protocol.

timeout	integer	Keep alive timeout for the persistent TLS protocol. Value range 10-3600 seconds, default value is 30 seconds. Only used for persistent TLS protocol.
sendSipsOverTLS	boolean	„1” or “true”, if send secure SIP over TLS is set. Only used for TLS or persistent TLS protocol.
authenticationTLS	boolean	„1” or “true”, if authentication for TLS is set. Only used for TLS or persistent TLS protocol.
commonNameValidationTLS	boolean	„1” or “true”, if common name validation for TLS is set. Only used for TLS or persistent TLS protocol. Ignored when authenticationTLS is not set.

Available filters: (none)

4.34.13 GetBackupSIP

With this request the client can query the SIP backup proxy/registrar settings.

Following fields are defined for the request *GetBackupSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetBackupSIPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
failoverActive	boolean	yes	„1” or “true”, if an automatic failover to secondary/tertiary servers or automatic coming back to primary servers shall be performed.
failoverTime	integer	yes	Failover keep alive time in minutes. Range 5 to 60 minutes
secondaryProxyServer	string	yes	Proxy server name or IP address of the secondary SIP proxy server
secondaryProxyPort	integer	yes	Proxy server port of the secondary SIP proxy server
secondaryRegServer	string	yes	Registrar server name or IP address of the secondary SIP registrar server
secondaryRegPort	integer	yes	Registrar server port of the secondary SIP registrar server
secondaryOutboundProxyServer	string	yes	Outbound proxy server name or IP address of the

			secondary SIP outbound proxy server
secondaryOutboundProxyPort	integer	yes	Outbound proxy server port of the secondary SIP outbound proxy server
tertiaryProxyServer	string	yes	Proxy server name or IP address of the tertiary SIP proxy server
tertiaryProxyPort	integer	yes	Proxy server port of the tertiary SIP proxy server
tertiaryRegServer	string	yes	Registrar server name or IP address of the tertiary SIP registrar server
tertiaryRegPort	integer	yes	Registrar server port of the tertiary SIP registrar server
tertiaryOutboundProxyServer	string	yes	Outbound proxy server name or IP address of the tertiary SIP outbound proxy server
tertiaryOutboundProxyPort	integer	yes	Outbound proxy server port of the tertiary SIP outbound proxy server

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.34.14 SetBackupSIP

With this request the client can set the SIP backup proxy/registrar settings.

Following fields are defined for the request *SetBackupSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
failoverActive	boolean	no	„1” or “true”, if an automatic failover to secondary/tertiary servers or automatic coming back to primary servers shall be performed. Default=false
failoverTime	integer	no	Failover keep alive time in minutes. Range 5 to 60 minutes (default 10 min.)
secondaryProxyServer	string	no	Proxy server name or IP address of the secondary SIP proxy server
secondaryProxyPort	integer	no	Proxy Server port of the secondary SIP proxy server
secondaryRegServer	string	no	Registrar Server name or IP address of the secondary SIP registrar server
secondaryRegPort	integer	no	Registrar Server port of the secondary SIP registrar server
secondaryOutboundProxyServer	string	no	Outbound proxy server name or IP address of the secondary SIP outbound proxy server
secondaryOutboundProxyPort	integer	no	Outbound proxy server port of the secondary SIP

			outbound proxy server
tertiaryProxyServer	string	no	Proxy server name or IP address of the tertiary SIP proxy server
tertiaryProxyPort	integer	no	Proxy server port of the tertiary SIP proxy server
tertiaryRegServer	string	no	Registrar server name or IP address of the tertiary SIP registrar server
tertiaryRegPort	integer	no	Registrar server port of the tertiary SIP registrar server
tertiaryOutboundProxyServer	string	no	Outbound proxy server name or IP address of the tertiary SIP outbound proxy server
tertiaryOutboundProxyPort	integer	no	Outbound proxy server port of the tertiary SIP outbound proxy server

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetBackupSIPResp*. It contains all actual set attributes, all attributes are optional.

When this request succeeds and at least one server configuration changes, the OMM may reboot automatically.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.34.15 EventBackupSIPCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
failoverActive	boolean	no	„1” or “true”, if an automatic failover to secondary/tertiary servers or automatic coming back to primary servers shall be performed.
failoverTime	integer	no	Failover keep alive time in minutes, if changed. Range 5 to 60 minutes (default 10 min.)
secondaryProxyServer	string	no	Proxy server name or IP address of the secondary SIP proxy server, if changed
secondaryProxyPort	integer	no	Proxy server port of the secondary SIP proxy server, if changed
secondaryRegServer	string	no	Registrar server name or IP address of the secondary SIP registrar server, if changed

secondaryRegPort	integer	no	Registrar server port of the secondary SIP registrar server, if changed
secondaryOutboundProxyServer	string	no	Outbound proxy server name or IP address of the secondary SIP outbound proxy server, if changed
secondaryOutboundProxyPort	integer	no	Outbound proxy server port of the secondary SIP outbound proxy server, if changed
tertiaryProxyServer	string	no	Proxy server name or IP address of the tertiary SIP proxy server, if changed
tertiaryProxyPort	integer	no	Proxy server port of the tertiary SIP proxy server, if changed
tertiaryRegServer	string	no	Registrar server name or IP address of the tertiary SIP registrar server, if changed
tertiaryRegPort	integer	no	Registrar server port of the tertiary SIP registrar server, if changed
tertiaryOutboundProxyServer	string	no	Outbound proxy server name or IP address of the tertiary SIP outbound proxy server, if changed
tertiaryOutboundProxyPort	integer	no	Outbound proxy server port of the tertiary SIP outbound proxy server, if changed

Available filters: (none)

4.34.16 GetRTP

With this request the client can query the RTP settings.

Following fields are defined for the request *GetRTP* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetRTPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
portBase	integer	yes	RTP port base
packetTime	integer	yes	RTP packet size in milliseconds
silenceSupp	boolean	yes	„1” or “true”, for silence suppression
receiverPrecedence	boolean	yes	„1” or “true”, for receiver precedence on CODEC negotiation
comfortNoisePktElim	boolean	no	„1” or “true”, for eliminate comfort noise packets in media streams
singleCodecReplyInSDP	boolean	no	„1” or “true”, for reply an incoming SDP (session description)

			protocol) offer with a single audio codec in the SDP answer
--	--	--	---

Additionally this response contains a sequence of elements called *codec*. This is the list of preferred codecs. The list may also contain entries with value *None*. These entries can be used as a place holder. Each of these elements has following attribute:

Name	Type	Mandatory	Description
type	CodecType	yes	Codec type

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.34.17 SetRTP

With this request the client can set the RTP settings.

Following fields are defined for the request *SetRTP* in addition to the common attributes:

Name	Type	Mandatory	Description
portBase	integer	no	RTP port base
packetTime	integer	no	RTP packet size in milliseconds
silenceSupp	boolean	no	„1” or “true”, for silence suppression
receiverPrecedence	boolean	no	„1” or “true”, for receiver precedence on CODEC negotiation
comfortNoisePktElim	boolean, default=false	no	„1” or “true”, for eliminate comfort noise packets in media streams
singleCodecReplyInSDP	boolean	no	„1” or “true”, for reply an incoming SDP (session description protocol) offer with a single audio codec in the SDP answer

Additionally this request contains a sequence of elements called *codec*. This is the list of preferred codecs. Each of these elements has following attribute:

Name	Type	Mandatory	Description
type	CodecType	yes	Codec type

All attributes which have to be changed must be filled in by the client.

If the sequence of codecs is empty, the codec list is not changed. It is only replaced by a new list if the sequence contains at least one entry. The list may have up to 5 entries, spare entries are ignored. The list may also contain entries with value *None*. These entries can be used as a place holder.

The reply is an object called *SetRTPResp*. It contains all actual set attributes, all attributes are optional.

When this request succeeds, the OMM may reboot automatically.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.34.18 EventRTPCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
portBase	integer	no	RTP port base
packetTime	integer	no	RTP packet size in milliseconds
silenceSupp	boolean	no	„1” or “true”, for silence suppression
receiverPrecedence	boolean	no	„1” or “true”, for receiver precedence on CODEC negotiation
comfortNoisePktElim	boolean	no	„1” or “true”, for eliminate comfort noise packets in media streams
singleCodecReplyInSDP	boolean	no	„1” or “true”, for reply an incoming SDP (session description protocol) offer with a single audio codec in the SDP answer

Additionally this event may contain an updated sequence of elements called *codec*. This is the list of preferred codecs. If the list is not sent in the event, it has not been changed. Each of these elements has following attribute:

Name	Type	Mandatory	Description
type	CodecType	no	Codec type

Available filters: (none)

4.34.19 GetDTMF

With this request the client can query the DTMF settings.

Following fields are defined for the request *GetDTMF* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetDTMFResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
outOfBand	boolean	yes	„1” or “true”, for out-of-band DTMF
payloadType	integer	yes	DTMF payload type

method	DTMFMethodType	yes	DTMF out-of-band method
--------	----------------	-----	-------------------------

DTMFMethodType is an enumeration with one of these values:

Value	Description
RFC2833	DTMF according to RFC 2833
INFO	DTMF in SIP INFO
Both	RFC 2833 and SIP INFO

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.34.20 SetDTMF

With this request the client can set the DTMF settings.

Following fields are defined for the request *SetDTMF* in addition to the common attributes:

Name	Type	Mandatory	Description
outOfBand	boolean	no	„1” or “true”, for out-of-band DTMF
payloadType	integer	no	DTMF payload type
method	DTMFMethodType	no	DTMF out-of-band method

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetDTMFResp*. It contains all actual set attributes, all attributes are optional.

When this request succeeds, the OMM may reboot automatically.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.34.21 EventDTMFCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
outOfBand	boolean	no	„1” or “true”, for out-of-band DTMF, if changed
payloadType	integer	no	DTMF payload type, if changed

method	DTMFMethodType	no	DTMF out-of-band method, if changed
--------	----------------	----	-------------------------------------

Available filters: (none)

4.34.22 GetRegistrationTrafficShaping

With this request the client can query the Registration Traffic Shaping settings.

Following fields are defined for the request *GetRegistrationTrafficShaping* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetRegistrationTrafficShapingResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
maxRegistrations	integer	yes	The maximum number simultaneously started registrations. Allowed range: 1 - 100
timeout	integer	yes	The waiting time between a registration finish and starting the next registration in ms. Allowed range: 0 – 1000 msec
spreadRegRenew	boolean	yes	Spread registration renewals. If enabled the OMM spreads automatically the registration renewals of all phones between half-way registration period and 30 seconds prior to the expiration. This prevents bulks of registration renewals.
renewalTimer	integer	yes	If <i>spreadRegRenew</i> is disabled the <i>renewalTimer</i> is the value, in seconds, prior to expiration, that the OMM renews registrations. The phone will automatically send registration renewals half-way through the registration period, unless half-way is more than the threshold value.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.34.23 SetRegistrationTrafficShaping

With this request the client can set the Registration Traffic Shaping settings.

Following fields are defined for the request *SetRegistrationTrafficShaping* in addition to the common attributes:

Name	Type	Mandatory	Description
maxRegistrations	integer	no	The maximum number simultaneously started registrations. Allowed range: 1 - 100
timeout	integer	no	The waiting time between a registration finish and starting the next registration in msec. Allowed range: 0 – 1000 msec
spreadRegRenew	boolean	no	Spread registration renewals. If enabled the OMM spreads automatically the registration renewals of all phones between half-way registration period and 30 seconds prior

			to the expiration. This prevents bulks of registration renewals.
renewalTimer	integer	no	If <i>spreadRegRenew</i> is disabled the <i>renewalTimer</i> is the value, in seconds, prior to expiration, that the OMM renews registrations. The phone will automatically send registration renewals half-way through the registrationperiod, unless half-way is more than the threshold value.

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetRegistrationTrafficShapingResp*. It contains all actual set attributes, all attributes are optional.

When this request succeeds, the OMM may reboot automatically.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.34.24 EventRegistrationTrafficShapingCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
maxRegistrations	integer	no	The maximum number simultaneously started registrations. Allowed range: 1 - 100
timeout	integer	no	The waiting time between a registration finish and starting the next registration in ms. Allowed range: 0 – 1000 ms
spreadRegRenew	boolean	no	Spread registration renewals. If enabled the OMM spreads automatically the registration renewals of alls phones between half-way registration period and 30 seconds prior to the expiration. This prevents bulks of registration renewals.
renewalTimer	integer	no	If <i>spreadRegRenew</i> is disabled the <i>renewalTimer</i> is the value, in seconds, prior to expiration, that the OMM renews registrations. The phone will automatically send registration renewals half-way through the registrationperiod, unless half-way is more than the threshold value.

Available filters: (none)

4.34.25 GetSuplServ

With this request the client can query the SIP supplementary service (SuplServ) settings.

Following fields are defined for the request *GetSuplServ* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetSuplServResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
callForwDiv	boolean	yes	„1” or “true”, if call forwarding /diversion is activated.
locLineHndlg	boolean	yes	„1” or “true”, if local line handling is activated.
callTransferByHook	boolean	yes	„1” or “true”, if in brokering or inquiry call states, the 142d DECT phone device user can invoke a call transfer by pressing the hook key.
uriSeparator	boolean	yes	„1” or “true”, if the user name in a SIP URI (in to/from header or p-asserted-identity) is separated by a ‘;’, the following prefix will be cut and not printed to call logs or call display of the DECT phone which receives that SIP message. Example: “SIP URI” sip:4711:postd=pp22@foo.com:user=phone If uriSeparator=true, just “4711” will be printed Else “4711;postd=pp22” will be printed
reRegisterAfterFailOver	boolean	yes	„1” or “true”, if the a SIP registering has to be performed after OMM standby fail over with becoming both OMMs active in a network broken into two parts. When both OMMs get resynced with each other SIP registration has to be performed again.
ringingOnHold	boolean	yes	„1” or “true”, if ringback on hold is signaled with ringing instead of an inband tone only.
transferByHook6xxd	boolean	yes	„1” or “true”, if call transfer during hook on to the held call partner is initiated. If “0” or “false”, callback is initiated to the held partner. Only available for 6xxd DECT phones.
releaseInfoTimerActiveCall	integer	yes	Release info timeout in seconds for an active call. Default is 5 seconds.
releaseInfoTimerFailedCall	integer	yes	Release info timeout in seconds for a failed call. Default is 5 seconds.
releaseInfoTimerHoldCall	integer	yes	Release info timeout in seconds for a held call. Default is 5 seconds.

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: AllCnfRead

4.34.26 SetSuplServ

With this request the client can set the SIP supplementary service (SuplServ) settings.

Following fields are defined for the request *SetSup/Serv* in addition to the common attributes:

Name	Type	Mandatory	Description
callForwDiv	boolean	no	„1” or “true”, if call forwarding /diversion is activated.
locLineHndlg	boolean	no	„1” or “true”, if local line handling is activated.
callTransferByHook	boolean	no	„1” or “true”, if in brokering or inquiry call states, the 142d DECT phone device user can invoke a call transfer by pressing the hook key.
uriSeparator	boolean	no	„1” or “true”, if the user name in a SIP URI (in to/from header or p-asserted-identity) is separated by a ‘;’, the following prefix will be cut and not printed to call logs or call display of the DECT phone which receives that SIP message. Example: “SIP URI” sip:4711;postd=pp22@foo.com:user=phone If uriSeparator=true, just “4711” will be printed Else “4711;postd=pp22” will be printed
reRegisterAfterFailOver	boolean	no	„1” or “true”, if the a SIP registering has to be performed after OMM standby fail over with becoming both OMMs active in a network broken into two parts. When both OMMs get resynced with each other SIP registration has to be performed again.
ringingOnHold	boolean	no	„1” or “true”, if ringback on hold is signaled with ringing instead of an inband tone only.
transferByHook6xxd	boolean	no	„1” or “true”, if call transfer during hook on to the held call partner is initiated. If “0” or “false”, callback is initiated to the held partner. Only available for 6xxd DECT phones.
releaseInfoTimerActiveCall	integer	no	Release info timeout in seconds for an active call. Default is 5 seconds.
releaseInfoTimerFailedCall	integer	no	Release info timeout in seconds for a failed call. Default is 5 seconds.
releaseInfoTimerHoldCall	integer	no	Release info timeout in seconds for a held call. Default is 5 seconds.

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetSup/ServResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: AIIcnfWrite

4.34.27 EventSuplServCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
callForwDiv	boolean	no	„1” or “true”, if call forwarding /diversion is activated.
locLineHndlg	boolean	no	„1” or “true”, if local line handling is activated.
callTransferByHook	boolean	no	„1” or “true”, if in brokering or inquiry call states, the 142d DECT phone device user can invoke a call transfer by pressing the hook key.
uriSeparator	boolean	no	„1” or “true”, if the user name in a SIP URI (in to/from header or p-asserted-identity) is separated by a ‘;’, the following prefix will be cut and not printed to call logs or call display of the DECT phone which receives that SIP message. Example: “SIP URI” sip:4711;postd=pp22@foo.com;user=phone If uriSeparator=true, just “4711” will be printed Else “4711;postd=pp22” will be printed
reRegisterAfterFailOver	boolean	no	„1” or “true”, if the a SIP registering has to be performed after OMM standby fail over with becoming both OMMs active in a network broken into two parts. When both OMMs get resynced with each other SIP registration has to be performed again.
ringingOnHold	boolean	no	„1” or “true”, if ringback on hold is signaled with ringing instead of an inband tone only.
transferByHook6xxd	boolean	no	„1” or “true”, if call transfer during hook on to the held call partner is initiated. If “0” or “false”, callback is initiated to the held partner. Only available for 6xxd DECT phones.
releaseInfoTimerActiveCall	integer	no	Release info timeout in seconds for an active call. Default is 5 seconds.
releaseInfoTimerFailedCall	integer	no	Release info timeout in seconds for a failed call. Default is 5 seconds.
releaseInfoTimerHoldCall	integer	no	Release info timeout in seconds for a held call. Default is 5 seconds.

Available filters: (none)

4.34.28 GetConferenceServerSIP

With this request the client can query the SIP conference settings.

Following fields are defined for the request *GetConferenceServerSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetConferenceServerSIPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
conferenceServerType	string	yes	Type of the global conference server. One of "None", "External", "ExternalBlindTransfer" or "Integrated".
conferenceServerURI	string	yes	URI for the global conference server

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.34.29 SetConferenceServerSIP

With this request the client can set the SIP conference settings.

Following fields are defined for the request *SetConferenceServerSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
conferenceServerType	string	no	Type of the global conference server. One of "None", "External", "ExternalBlindTransfer" or "Integrated".
conferenceServerURI	string	no	URI for the global conference server

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetConferenceServerSIPResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.34.30 EventConferenceServerSIPCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf* or *ConferenceCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
conferenceServerType	string	no	Type of the global conference server. One of "None", "External", "ExternalBlindTransfer" or "Integrated".

conferenceServerURI	string	no	URI for the global conference server
---------------------	--------	----	--------------------------------------

Available filters: (none)

4.34.31 GetPortRangeSIP

With this request the client can query the SIP port range settings.

Following fields are defined for the request *GetPortRangeSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetPortRangeSIPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
userUdpTcp	SIPPortRangeType	yes	UDP/TCP SIP port range for registering users at the call server. Default is a range from port 5060 to 5060.
userTls	SIPPortRangeType	yes	TLS SIP port range for registering users at the call server. Default is a range from port 5061 to 5061.
confRoomUdpTcp	SIPPortRangeType	yes	UDP/TCP SIP port range for registering conference rooms at the call server. Default is a range from port 5062 to 5062.
confRoomTls	SIPPortRangeType	yes	TLS SIP port range for registering conference rooms at the call server. Default is a range from port 5063 to 5063.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AiICnfRead*

4.34.32 SetPortRangeSIP

With this request the client can set the SIP port range settings.

Following fields are defined for the request *SetPortRangeSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
userUdpTcp	SIPPortRangeType	no	UDP/TCP SIP port range for registering users at the call server. Default is a range from port 5060 to 5060.
userTls	SIPPortRangeType	no	TLS SIP port range for registering users at the call server. Default is a range from port 5061 to 5061.
confRoomUdpTcp	SIPPortRangeType	no	UDP/TCP SIP port range for registering conference rooms at the call server. Default is a range from port 5062 to 5062.

confRoomTls	SIPPortRangeType	no	TLS SIP port range for registering conference rooms at the call server. Default is a range from port 5063 to 5063.
-------------	------------------	----	--

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetPortRangeSIPResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.34.33 EventPortRangeSIPCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
userUdpTcp	SIPPortRangeType	no	UDP/TCP SIP port range for registering users at the call server. Default is a range from port 5060 to 5060.
userTls	SIPPortRangeType	no	TLS SIP port range for registering users at the call server. Default is a range from port 5061 to 5061.
confRoomUdpTcp	SIPPortRangeType	no	UDP/TCP SIP port range for registering conference rooms at the call server. Default is a range from port 5062 to 5062.
confRoomTls	SIPPortRangeType	no	TLS SIP port range for registering conference rooms at the call server. Default is a range from port 5063 to 5063.

Available filters: (none)

4.34.34 GetIntercomCallHandlingSIP

With this request the client can query the SIP intercom call handling settings.

Following fields are defined for the request *GetIntercomCallHandlingSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetIntercomCallHandlingSIPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

autoAnswer	boolean	yes	„1” or “true”, if auto-answer global setting for incoming calls is activated. If auto-answer is enabled, the DECT phone plays a tone to alert the user before answering the call. If auto-answer is disabled, the DECT phone treats the incoming call as a normal call. See also the user individual setting of <i>autoAnswer</i> in PPUserType (4.39.1).
microphoneMute	boolean	yes	„1” or “true”, if microphone mute global setting for incoming calls is activated. Enables or disables the microphone on the DECT phone for calls made by the originating caller. See also the user individual setting of <i>microphoneMute</i> in PPUserType (4.39.1).
warningTone	boolean	yes	„1” or “true”, if warning tone global setting for incoming calls is activated. Enables or disables a warning tone to play when the DECT phone receives an incoming call on an active line. See also the user individual setting of <i>warningTone</i> in PPUserType (4.39.1).
allowBargeIn	boolean	yes	„1” or “true”, if allow barge in global setting for incoming calls is activated. Enables or disables how the DECT phone handles incoming calls while the DECT phone is on an active call. When enabled an incoming call takes precedence over an active call, by placing the active call on hold and automatically answering the call. When disabled the DECT phone treats an incoming call like a normal call. See also the user individual setting of <i>allowBargeIn</i> in PPUserType (4.39.1).
pushToTalkPrefix	string	yes	The OMM call control will proceed that received SIP events, if they fit to the configuration referring to audio and auto answer settings. This feature will only be available for 6xxd DECT phones, version 4.0 or higher.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.34.35 SetIntercomCallHandlingSIP

With this request the client can set the SIP intercom call handling settings.

Following fields are defined for the request *SetIntercomCallHandlingSIP* in addition to the common attributes:

Name	Type	Mandatory	Description
autoAnswer	boolean	no	„1” or “true”, if auto-answer global setting for incoming calls is activated. If auto-answer is enabled, the DECT phone plays a tone to alert the user before answering the call. If auto-answer is disabled, the DECT phone treats the incoming call as a normal call. See also the user individual setting of <i>autoAnswer</i> in PPUserType (4.39.1).
microphoneMute	boolean	no	„1” or “true”, if microphone mute global setting for incoming calls is activated.

			Enables or disables the microphone on the DECT phone for calls made by the originating caller. See also the user individual setting of <i>microphoneMute</i> in PPUserType (4.39.1).
warningTone	boolean	no	„1” or “true”, if warning tone global setting for incoming calls is activated. Enables or disables a warning tone to play when the DECT phone receives an incoming call on an active line. See also the user individual setting of <i>warningTone</i> in PPUserType (4.39.1).
allowBargeIn	boolean	no	„1” or “true”, if allow barge in global setting for incoming calls is activated. Enables or disables how the DECT phone handles incoming calls while the DECT phone is on an active call When enabled an incoming call takes precedence over an active call, by placing the active call on hold and automatically answering the call. When disabled the DECT phone treats an incoming call like a normal call. See also the user individual setting of <i>allowBargeIn</i> in PPUserType (4.39.1).
pushToTalkPrefix	string	yes	The OMM call control will proceed that received SIP events, if they fit to the configuration referring to audio and auto answer settings. This feature will only be available for 6xxd DECT phones, version 4.0 or higher.

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetIntercomCallHandlingSIPResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.34.36 EventIntercomCallHandlingSIPCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
autoAnswer	boolean	no	„1” or “true”, if auto-answer global setting for incoming calls is activated. If auto-answer is enabled, the DECT phone plays a tone to alert the user before answering the call. If auto-answer is disabled, the DECT phone treats the incoming call as a normal call. See also the user individual setting of <i>autoAnswer</i> in PPUserType (4.39.1).

microphoneMute	boolean	no	„1” or “true”, if microphone mute global setting for incoming calls is activated. Enables or disables the microphone on the DECT phone for calls made by the originating caller. See also the user individual setting of <i>microphoneMute</i> in PPUserType (4.39.1).
warningTone	boolean	no	„1” or “true”, if warning tone global setting for incoming calls is activated. Enables or disables a warning tone to play when the DECT phone receives an incoming call on an active line. See also the user individual setting of <i>warningTone</i> in PPUserType (4.39.1).
allowBargeIn	boolean	no	„1” or “true”, if allow barge in global setting for incoming calls is activated. Enables or disables how the DECT phone handles incoming calls while the DECT phone is on an active call When enabled an incoming call takes precedence over an active call, by placing the active call on hold and automatically answering the call. When disabled the DECT phone treats an incoming call like a normal call. See also the user individual setting of <i>allowBargeIn</i> in PPUserType (4.39.1).
pushToTalkPrefix	string	yes	The OMM call control will proceed that received SIP events, if they fit to the configuration referring to audio and auto answer settings. This feature will only be available for 6xxd DECT phones, version 4.0 or higher.

Available filters: (none)

4.35 Secure Certificate Configuration And Maintenance

4.35.1 Secure Provisioning Certificate Handling

Used for a secure OMM interaction with the provisioning server.

4.35.1.1 GetSecurePROVCertificate

This is not a valid AXI request. The provisioning certificates are part of the configuration URL (see 4.22.7).

4.35.1.2 SetSecurePROVCertificate

This is not a valid AXI request. The provisioning certificates are part of the configuration URL (see 4.22.8).

4.35.1.3 EventSecurePROVCertificateCnf

This is not a valid AXI event. The provisioning certificates are part of the configuration URL (see 4.22.9).

4.35.1.4 GetSecurePROVCertificateServerImport

With this request the client can query the secure provisioning certificate server import settings.

Following fields are defined for the request *GetSecurePROVCertificateServerImport* in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

			(no additional fields defined)
--	--	--	--------------------------------

The reply is an object called *GetSecurePROVCertificateServerImportResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Secure provisioning certificate server URL to read the certificates from.
trustedCertificates	string	yes	Filename of the trusted certificates to read from the server.
localCertificates	string	yes	Filename of the local certificates to read from the server.
privateKeys	string	yes	Filename of the private key certificate to read from the server.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.35.1.5 SetSecurePROVCertificateServerImport

With this request the client can set the secure provisioning certificate server import settings.

Following fields are defined for the request *SetSecurePROVCertificateServerImport* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	no	Secure provisioning certificate server URL to read the certificates from.
trustedCertificates	string	no	Filename of the trusted certificates to read from the server.
localCertificates	string	no	Filename of the local certificates to read from the server.
privateKeys	string	no	Filename of the private key certificate to read from the server.

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetSecurePROVCertificateServerImportResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Secure provisioning certificate server URL to read the certificates from.
trustedCertificates	string	yes	Filename of the trusted certificates to read from the server.
localCertificates	string	yes	Filename of the local certificates to read from the server.

privateKeys	string	yes	Filename of the private key certificate to read from the server.
-------------	--------	-----	--

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.35.1.6 EventSecurePROVCertificateServerImportCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	no	Secure provisioning certificate server URL to read the certificates from.
trustedCertificates	string	no	Filename of the trusted certificates to read from the server.
localCertificates	string	no	Filename of the local certificates to read from the server.
privateKeys	string	no	Filename of the private key certificate to read from the server.

Available filters: (none)

4.35.2 Secure OMM Certificate Handling

Used for a secure OMM client interaction with the OMM WEB service or OMM AXI interface.

4.35.2.1 GetSecureOMMCertificate

With this request the client can query the secure OMM certificate settings.

Following fields are defined for the request *GetSecureOMMCertificate* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetSecureOMMCertificateResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
nLocalCertificates	integer	yes	Number of set local certificate chains.

nPrivateKeys	integer	yes	Number of set private keys.
localCertificates	PemCertificateListType	no	The local certificate chain settings.
privateKeys	PemCertificateListType	no	The set OMM private key set. This key is used for persistent TLS only if the server verifies the client certificate.
privateKeyPassword	string	no	Optional password for the private key certificate access, encrypted with public key. If not set this string is empty!

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

Note, that this request contains security risks! Therefore the response contains the *privateKey* sent in an asymmetrically encrypted form only when the request is done from the ICS client. The *localCertificates* and *trustedCertificates* are not encrypted and sent to all clients.

The version info is represented by the element *GetOMMCertificate*.

4.35.2.2 GetOMMCertificate

This request is identical to *GetSecureOMMCertificate*.

4.35.2.3 SetSecureOMMCertificate

With this request the client can set the secure OMM certificate settings.

Following fields are defined for the request *SetSecureOMMCertificate* in addition to the common attributes:

Name	Type	Mandatory	Description
localCertificates	PemCertificateListType	no	All local certificate chains to set. Before setting the new local certificate chains in the OMM all old local certificate chain settings will be deleted. If this element is missing, the local certificate chains stay unchanged.
privateKeys	PemCertificateListType	no	One OMM private key to set. This key is used for persistent TLS only if the server verifies the client certificate. If this element is missing, the private key stays unchanged.
privateKeyPassword	string	no	Optional password for the private key certificate access, encrypted with public key.

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetSecureOMMCertificateResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
nLocalCertificates	integer	yes	Number of set local certificate chains.

nPrivateKeys	integer	yes	Number of set private keys.
privateKeyPassword	string	yes	Optional password for the private key certificate access, encrypted with public key.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.35.2.4 SetOMMCertificate

This request is identical to *SetSecureOMMCertificate*.

4.35.2.5 EventOMMCertificateCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
nLocalCertificates	integer	yes	Number of set local certificate chains.
nPrivateKeys	integer	yes	Number of set private keys.
localCertificates	PemCertificateListType	no	All local certificate chains set. Before setting the new local certificate chains in the OMM all old local certificate chain settings will be deleted.
privateKeys	PemCertificateListType	no	The set OMM private key set. This key is used for persistent TLS only if the server verifies the client certificate.
privateKeyPassword	string	no	Optional password for the private key certificate access, encrypted with public key. If not set this string is empty!

Available filters: (none)

Note, that this event contains security risks! Therefore the event contains the *privateKey* sent in an asymmetrically encrypted form only when the subscription is done from the ICS client. The *localCertificates* is not encrypted and notified to all clients.

The version info is represented by the element *SetOMMCertificate*.

4.35.2.6 GetSecureOMMCertificateServerImport

With this request the client can query the secure OMM certificate server import settings.

Following fields are defined for the request *GetSecureOMMCertificateServerImport* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetSecureOMMCertificateServerImportResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Secure OMM certificate server URL to read the certificates from.
localCertificates	string	yes	Filename of the local certificates to read from the server.
privateKeys	string	yes	Filename of the private key certificate to read from the server.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.35.2.7 SetSecureOMMCertificateServerImport

With this request the client can set the secure OMM certificate server import settings.

Following fields are defined for the request *SetSecureOMMCertificateServerImport* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	no	Secure OMM certificate server URL to read the certificates from.
localCertificates	string	no	Filename of the local certificates to read from the server.
privateKeys	string	no	Filename of the private key certificate to read from the server.

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetSecureOMMCertificateServerImportResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Secure OMM certificate server URL to read the certificates from.
localCertificates	string	yes	Filename of the local certificates to read from the server.
privateKeys	string	yes	Filename of the private key certificate to read from

			the server.
--	--	--	-------------

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.35.2.8 EventSecureOMMCertificateServerImportCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	no	Secure OMM certificate server URL to read the certificates from.
localCertificates	string	no	Filename of the local certificates to read from the server.
privateKeys	string	no	Filename of the private key certificate to read from the server.

Available filters: (none)

4.35.3 Secure SIP Certificate Handling

Used for a secure OMM interaction with the SIP call server.

4.35.3.1 GetSecureSIPCertificate

With this request the client can query the secure SIP certificate settings.

Following fields are defined for the request *GetSecureSIPCertificate* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetSecureSIPCertificateResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
nTrustedCertificates	integer	yes	Number of set trusted certificates.
nLocalCertificates	integer	yes	Number of set local certificate chains.
nPrivateKeys	integer	yes	Number of set private keys.
trustedCertificates	PemCertificateListType	no	All trusted certificates settings.

localCertificates	PemCertificateListType	no	The local certificate chain settings.
privateKeys	PemCertificateListType	no	The set OMM private key set. This key is used for persistent TLS only if the server verifies the client certificate.
privateKeyPassword	string	no	Optional password for the private key certificate access, encrypted with public key. If not set this string is empty!

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

Note, that this request contains security risks! Therefore the response contains the *privateKey* sent in an asymmetrically encrypted form only when the request is done from the ICS client. The *localCertificates* and *trustedCertificates* are not encrypted and sent to all clients.

4.35.3.2 SetSecureSIPCertificate

With this request the client can set the secure SIP certificate settings.

Following fields are defined for the request *SetSecureSIPCertificate* in addition to the common attributes:

Name	Type	Mandatory	Description
trustedCertificates	PemCertificateListType	no	All trusted certificates to set. Before setting the new trusted certificates in the OMM all old trusted certificate settings will be deleted. If this element is missing, the trusted certificates stay unchanged.
localCertificates	PemCertificateListType	no	All local certificate chains to set. Before setting the new local certificate chains in the OMM all old local certificate chain settings will be deleted. If this element is missing, the local certificate chains stay unchanged.
privateKeys	PemCertificateListType	no	One OMM private key to set. This key is used for persistent TLS only if the server verifies the client certificate. If this element is missing, the private key stays unchanged.
privateKeyPassword	string	no	Optional password for the private key certificate access, encrypted with public key.

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetSecureSIPCertificateResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
nTrustedCertificates	integer	yes	Number of set trusted certificates.
nLocalCertificates	integer	yes	Number of set local certificate chains.

nPrivateKeys	integer	yes	Number of set private keys.
privateKeyPassword	string	yes	Optional password for the private key certificate access, encrypted with public key.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.35.3.3 EventSecureSIPCertificateCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
nTrustedCertificates	integer	yes	Number of set trusted certificates.
nLocalCertificates	integer	yes	Number of set local certificate chains.
nPrivateKeys	integer	yes	Number of set private keys.
trustedCertificates	PemCertificateListType	no	All trusted certificates set. Before setting the new trusted certificates in the OMM all old trusted certificate settings will be deleted.
localCertificates	PemCertificateListType	no	All local certificate chains set. Before setting the new local certificate chains in the OMM all old local certificate chain settings will be deleted.
privateKeys	PemCertificateListType	no	The set OMM private key set. This key is used for persistent TLS only if the server verifies the client certificate.
privateKeyPassword	string	no	Optional password for the private key certificate access, encrypted with public key. If not set this string is empty!

Available filters: (none)

Note, that this event contains security risks! Therefore the event contains the *privateKey* sent in an asymmetrically encrypted form only when the subscription is done from the ICS client. The *localCertificates* and *trustedCertificates* are not encrypted and notified to all clients..

4.35.3.4 GetSecureSIPCertificateServerImport

With this request the client can query the secure SIP certificate server import settings.

Following fields are defined for the request *GetSecureSIPCertificateServerImport* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetSecureSIPCertificateServerImportResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Secure SIP certificate server URL to read the certificates from.
trustedCertificates	string	yes	Filename of the trusted certificates to read from the server.
localCertificates	string	yes	Filename of the local certificates to read from the server.
privateKeys	string	yes	Filename of the private key certificate to read from the server.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.35.3.5 SetSecureSIPCertificateServerImport

With this request the client can set the secure SIP certificate server import settings.

Following fields are defined for the request *SetSecureSIPCertificateServerImport* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	no	Secure SIP certificate server URL to read the certificates from.
trustedCertificates	string	no	Filename of the trusted certificates to read from the server.
localCertificates	string	no	Filename of the local certificates to read from the server.
privateKeys	string	no	Filename of the private key certificate to read from the server.

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetSecureSIPCertificateServerImportResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Secure SIP certificate server URL to read the certificates from.

trustedCertificates	string	yes	Filename of the trusted certificates to read from the server.
localCertificates	string	yes	Filename of the local certificates to read from the server.
privateKeys	string	yes	Filename of the private key certificate to read from the server.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.35.3.6 EventSecureSIPCertificateServerImportCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	no	Secure SIP certificate server URL to read the certificates from.
trustedCertificates	string	no	Filename of the trusted certificates to read from the server.
localCertificates	string	no	Filename of the local certificates to read from the server.
privateKeys	string	no	Filename of the private key certificate to read from the server.

Available filters: (none)

4.36 DECT Phone User Login Administration Configuration And Maintenance

4.36.1 GetPPLLoginVariant

With this request the client can query the state of DECT phone user login variant in case of unbound devices. During the login or logout procedure at the DECT phone the user can either be identified by its unique call number or by its unique additional id.

Following fields are defined for the request *GetPPLLoginVariant* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetPPLLoginVariantResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
login	enumeration	yes	One of “NUMBER” (default) [Login identification is done by the unique users call number / see 4.39.1] or “ID” [Login identification is done by the unique users additional id / see 4.39.1].

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.36.2 SetPPLLoginVariant

With this request the client can set the DECT phone login variant.

Following fields are defined for the request *SetPPLLoginVariant* in addition to the common attributes:

Name	Type	Mandatory	Description
login	enumeration	yes	One of “NUMBER” or “ID”.

The reply is an object called *SetPPLLoginVariantResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
login	enumeration	yes	One of “NUMBER” or “ID”.

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.36.3 EventPPLLoginVariantCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
login	enumeration	yes	One of “NUMBER” or “ID”.

Available filters: (none)

4.37 Restricted Subscription Duration Configuration And Maintenance

4.37.1 GetRestrictedSubscriptionDuration

With this request the client can query the state of restricted DECT phone subscription duration.

Following fields are defined for the request *GetRestrictedSubscriptionDuration* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetRestrictedSubscriptionDurationResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
restrictedSubscrDur	boolean	yes	„1” or “true”, if the subscription time to subscribe DECT phones to the system shall be restricted to only 2 minutes. Otherwise the subscription time is not limited at system start up or when subscription will be enabled. Default value is “false”.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.37.2 SetRestrictedSubscriptionDuration

With this request the client can set the restricted DECT phone subscription duration.

Following fields are defined for the request *SetRestrictedSubscriptionDuration* in addition to the common attributes:

Name	Type	Mandatory	Description
restrictedSubscrDur	boolean	yes	„1” or “true”, if the subscription time to subscribe DECT phones to the system shall be restricted to only 2 minutes. Otherwise the subscription time is not limited at system start up or when subscription will be enabled. Default value is “false”.

The reply is an object called *SetRestrictedSubscriptionDurationResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
restrictedSubscrDur	boolean	yes	„1” or “true”, if the subscription time to subscribe DECT phones to the system shall be restricted to only 2 minutes. Otherwise the subscription time is not limited at system start up or when subscription will be enabled. Default value is “false”.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.37.3 EventRestrictedSubscriptionDurationCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
restrictedSubscrDur	boolean	yes	„1” or “true”, if the subscription time to subscribe DECT phones to the system shall be restricted to only 2 minutes. Otherwise the subscription time is not limited at system start up or when subscription will be enabled. Default value is “false”.

Available filters: (none)

4.38 DECT Phone Device Configuration And Maintenance

4.38.1 PPDevType

This type contains all data fields of a DECT phone device. It is used in different requests and responses defined in this document. Not all fields are used in all OMM versions and in all request and response types.

Name	Type	Description
ppn	integer	Portable part number, unique ID for a DECT phone devices in the OMM. PPNs start at 1, i.e. 0 is an invalid PPN.
timeStamp	long integer	Last change time stamp of this element in seconds since 1970/1/1. When this element is not used in a create or set request the element is only changed when the data record differs from the record already stored in the OMM database,
relType	PPRelTypeType	Type or state of a relationship to a DECT phone user
uid	integer	User ID of the DECT phone user which is linked to this DECT phone device, if any.
ipei	string	IPEI, globally unique identifier for a DECT phone.
ac	string	Authentication code for DECT subscription
s	DECTSubscriptionStateType, default = "No"	DECT subscription state
uak	string	Encrypted UAK value
encrypt	boolean	„1” or “true”, if DECT encryption is enabled
capMessaging	boolean	„1” or “true”, if this device is known to support messaging (read only)
capMessagingForInternalUse	boolean	„1” or “true”, if this device is known to support messaging for internal use only (read only)
capEnhLocating	boolean	„1” or “true”, if this device is able to provide

		enhanced locating information (read only)
capBluetooth	boolean	„1” or “true”, if this device is able to use bluetooth (read only).
ethAddr	string	Bluetooth ethernet address, format “00:11:22:aa:bb:cc” (read only). An empty string means no Bluetooth equipment available at this device.
hwType	string	Detected device type. One of “142d”, “610d”, “620d”, “630d”, “612d”, “622d”, “632d”, “650c”, “740cv”, “GAP” or “Unknown” (read only).
ppProfileCapability	boolean	“1” or “true”, if the DECT phone is able to load DECT phone configuration data. This element is read only.
ppDefaultProfileLoaded	boolean	“1” or “true”, if the DECT phone has loaded the default DECT phone configuration data set. Note: If a user has logged in to this device additional user DECT phone configuration data might have been loaded to this DECT phone. In this case the value of this element is set to “0” or “false”. This element is read only.
subscribeToPARIOnly	boolean	“1” or “true”, if the DECT phone shall be subscribed only by using the PARI. This device cannot roam to other OMMs.

Which fields are mandatory depend on OMM type (refer to chapter 5), request and response type and permissions of the user being logged in.

PPRelType is an enumeration value which describes the type of a relation between a DECT phone device and a DECT phone user. It is defined as follows:

Value	Description
Unbound	This DECT phone device or DECT phone user can be bound dynamically, currently it is unbound.
Dynamic	This DECT phone device or DECT phone user can be bound dynamically, currently it is bound.
Fixed	This DECT phone device or DECT phone user has a fixed relation.

DECTSubscriptionState is an enumeration value defined as follows:

Value	Description
No	Not subscribed
Unconfirmed	The subscription was made but not confirmed by the DECT phone
Yes	The subscription was confirmed

4.38.2 GetPPDevSummary

This request is sent from the client to the OMM.

With this request the client can query common information about the DECT phone devices configured.

Following fields are defined for the request *GetPPDevSummary* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to this request is an object called *GetPPDevSummaryResp*. It contains following fields in addition to the common attributes:

Name	Type	Description
nRecords	integer	The total number of DECT phone devices set up.
ppnFirst	integer	The PPN of the first DECT phone set up.
subscribedDevs	integer	The total number of DECT subscribed devices/DECT phones

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.38.3 GetPPDev

With this request the client can acquire one or more DECT phone device data sets from the OMM. The request contains a PPN from which starting subsequent DECT phone device data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

PPNs usually ordered start at 1, but possibly not all numbers are assigned, i.e. the DECT phone device numbering may not be contiguous.

The responses are by PPN, starting with the smallest.

If the client asked for a certain number records but the given *ppn* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given PPN has to match to a record exactly. If there is a record with this PPN, it is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetPPDev* (*ppn* = "1", *maxRecords* = "20"). If the response contains 20 records, take the last PPN plus one and send another *GetPPDev*. Repeat this until you get less than 20 records back.

Following fields are defined for the request *GetPPDev* in addition to the common attributes:

Name	Type	Mandatory	Description
ppn	integer	yes	First PPN of DECT phone devices to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to *GetPPDev* is an object called *GetPPDevResp*. It contains following element in addition to the common attributes:

Name	Type	Description
pp	sequence of PPDevType	DECT phone device records, if found

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.38.4 CreatePPDev

The client can send this request to OM AXI to create a new DECT phone device data set. Refer to chapter 4.38.1 for information about which attributes of a DECT phone have to be set in a create request.

If there is no PPN assigned by the client, the OMM will choose one. If the client assigns a PPN, it must not exist yet.

Following element is defined for the request *CreatePPDev* in addition to the common attributes:

Name	Type	Mandatory	Description
pp	PPDevType	yes	Data of DECT phone to create.

Depending from the OMM type it may be allowed to create a DECT phone device without any predefined attribute. In this case the client may send an empty *CreatePP* request.

Newly created DECT phone devices must have the relation type *Unbound*, the device has to be subscribed already and has to have a valid UAK.

Note that writing *s="Yes"* or *s="Unconfirmed"* makes only sense when the data set has a valid UAK.

If the creation was successful, the reply contains all attributes which *GetPPDev* would return. Some data fields may be changed or set by the OMM. If the creation fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreatePPDevResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

Value	Description
ENoMem	No more DECT phone devices can be created

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.38.5 DeletePPDev

With this request the client can delete a DECT phone device record in the OMM. The key is its PPN.

Following fields are defined for the request *DeletePPDev* in addition to the common attributes:

Name	Type	Mandatory	Description
ppn	integer	yes	PPN of DECT phone device to be deleted

If the DECT phone device has a *Dynamic* relation to a DECT phone user they are unlinked automatically. If the device has a *Fixed* relation to a DECT phone user, both of them will be deleted.

The reply to this request is an object called *DeletePPDevResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

Possible values for *errCode*:

Value	Description
EPerm	No sufficient permissions for this request

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.38.6 SetPPDev

The client can send this request to OM AXI to change a DECT phone device data set. The PPN has to be filled in by the client to identify the record to be changed. Additionally the attributes which have to be changed must be filled in by the client. Refer to chapter 4.38.1 for information about which attributes of a DECT phone device may be changed using *SetPPDev*.

Following element is defined for the request *SetPPDev* in addition to the common attributes:

Name	Type	Mandatory	Description
pp	PPDevType	yes	Data of DECT phone device to be changed

If the change was successful, the reply contains all attributes which *GetPPDev* would return. Some data fields may be changed or set by the OMM. If the change fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The subscription state can only be changed to *No*. It cannot be changed at all when the device is *Unbound* or *Dynamic*.

It is not possible to change a relationship to a DECT phone user using this request. The attributes *relType* and *uid* are ignored. If a DECT phone device has the relation type *Dynamic* or *Unbound*, only the encryption flag can be changed.

The reply to this request is called *SetPPDevResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

Value	Description
EFailed	This device/attribute cannot be changed (e. g. because of relation type)

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.38.7 EventPPDevCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *PPDevCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with only a *ppn* in the *pp* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
pp	PPDevType	yes	Changed attributes
deleted	boolean, default false		„1” or “true”, if this data set has been deleted

Available filters: ppn

4.38.8 EventPPDevSummary

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf* or to *PPDevSummary*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes.

Attributes sent in this notification:

Name	Type	Description
nRecords	integer	The total number of DECT phone devices set up.
ppnFirst	integer	The PPN of the first DECT phone set up.
subscribedDevs	integer	The total number of DECT subscribed devices/DECT phones

4.38.9 EventPPTransaction

This event is sent by OM AXI when a DECT phone transaction took place. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to one of *PPTransaction* *PPTransactionLinkEstablish*, *PPTransactionRelease*, *PPTransactionEstablish*, *PPTransactionPPNotFound*, *PPTransactionPagingStarted*, *PPTransactionReleaseFromPP*, *PPTransactionPPSetupRejected*, *PPTransactionComsRelease*, *PPTransactionComsEstablish*, *PPTransactionSsFac*, *PPTransactionSsRelease*, *PPTransactionSsEstablish*, *PPTransactionConnHandover*, *PPTransactionLocReg*, *PPTransactionDetach*, or *PPCallStateRinging*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*, *Monitoring* or *Locating*.

The attribute *rfpld* is only sent when the *trType* provides this information and when the client does also have one of the permission *Monitoring* or *Locating*. This means that clients who only have the permission *AllCnfRead* get this event, but always without *rfpld*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
trType	TransactionTypeType	yes	One of the transaction types.

ppn	integer	yes	The DECT phone device related to this transaction.
rfpld	integer	no	Contains the ID of the RFP the event took place on. The presence of this parameter indicates also an established link to the DECT phone.

TransactionTypeType is an enumeration type which can have following values:

Value	Description
None	No transaction known (used in GetLastPPDevAction)
LinkEstablish	DECT link established (used in GetLastPPDevAction and EventPPTtransaction)
Release	Call released (used in GetLastPPDevAction and EventPPTtransaction)
Establish	Call established (used in GetLastPPDevAction and EventPPTtransaction)
PPNotFound	DECT phone not found (used in GetLastPPDevAction and EventPPTtransaction)
PagingStarted	Paging started (used in GetLastPPDevAction and EventPPTtransaction)
ReleaseFromPP	call released from DECT phone (used in GetLastPPDevAction and EventPPTtransaction)
PPSetupRejected	Call setup rejected (used in GetLastPPDevAction and EventPPTtransaction)
ComsRelease	Last connection oriented message service connection released (used in GetLastPPDevAction and EventPPTtransaction)
ComsEstablish	First connection oriented message service connection established (used in GetLastPPDevAction and EventPPTtransaction)
SsFac	Supplementary services facility code sent (used in GetLastPPDevAction and EventPPTtransaction)
SsRelease	Last supplementary services connection released (used in GetLastPPDevAction and EventPPTtransaction)
SsEstablish	First supplementary services connection established (used in GetLastPPDevAction and EventPPTtransaction)
ConnHandover	Connection hand over (used in GetLastPPDevAction and EventPPTtransaction)
LocReg	DECT phone location registration (used in GetLastPPDevAction and EventPPTtransaction)
Detach	DECT phone detached (used in GetLastPPDevAction and EventPPTtransaction)
Cc	Call control active (used in GetActivePPDev)
Ss	Supplementary services active (used in GetActivePPDev)
Coms	Connection oriented message service active (used in GetActivePPDev)

Available filters: ppn

4.38.10 GetLastPPDevAction

With this request the client can find out what is the last known DECT action of one or more DECT phone devices. The request contains a PPN from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

PPNs usually ordered start at 1, but possibly not all numbers are assigned, i.e. the DECT phone device numbering may not be contiguous.

The responses are by PPN, starting with the smallest.

If the client asked for a certain number records but there is no DECT phone device configured with the given *ppn*, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given PPN has to match to a record exactly. If there is a DECT phone device configured with this PPN, its last action is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetLastPPDevAction* (*ppn* = "1", *maxRecords* = "20"). If the response contains 20 records, take the last PPN plus one and send another *GetLastPPDevAction*. Repeat this until you get less than 20 records back.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
ppn	integer	yes	First PPN of DECT phone devices to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to this request is an object called *GetLastPPDevActionResp*. It contains following element in addition to the common attributes:

Name	Type	Description
pp	sequence of LastPPActionType	DECT phone action records, if found

The type LastPPActionType has following attributes:

Name	Type	Description
ppn	integer	PPN of the DECT phone
trType	TransactionTypeType	Last known transaction or "None"
rfpld	integer	RFP which was involved in this transaction, if applicable
relTime	integer	Relative time in seconds of the last transaction, e. g. "60" means that the action took place 60 seconds ago

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Monitoring, Locating*

4.38.11 GetActivePPDev

With this request the client can get one or more active DECT phone devices. The request contains a PPN from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

PPNs usually ordered start at 1, but possibly not all numbers are assigned, i.e. the DECT phone device numbering may not be contiguous.

The responses are by PPN, starting with the smallest.

If the client asked for a certain number records but there is no DECT phone device active with the given *ppn*, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given PPN has to match to a record exactly. If there is a DECT phone device active with this PPN, its last action is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetActivePPDev* (*ppn* = "1", *maxRecords* = "20"). If the response contains 20 records, take the last PPN plus one and send another *GetActivePPDev*. Repeat this until you get less than 20 records back.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
ppn	integer	yes	First PPN of DECT phone devices to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to this request is an object called *GetActivePPDevResp*. It contains following element in addition to the common attributes:

Name	Type	Description
pp	sequence of LastPPActionType	DECT phone action records for active DECT phones, if found

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.39 DECT Phone User Configuration And Maintenance

4.39.1 PPUserType

This type contains all data fields of a DECT phone user. It is used in different requests and responses defined in this document. Not all fields are used in all OMM versions and in all Request and Response types.

Name	Type	Description
uid	integer	User ID, numbering starts at 1, 0 is invalid
timeStamp	long integer	Last change time stamp of this element in seconds since 1970/1/1. When this element is not used in a create or set request the element is only changed when the data record differs from the record already stored in the OMM database,

relType	PPRelTypeType	Type or state of a relationship to a DECT phone device
ppn	integer	PPN of the DECT phone device which is linked to this DECT phone user, if any
name	string	User name See 4.64, if in case of setting the user name a truncation shall be allowed.
num	string	Phone number or SIP user ID. Also used as user credential account for the server authentication process to user configuration file on the server (when <i>login=Number</i> in <i>PPLginVariant</i>).
hierarchy1	string	Description hierarchy level 1, free format text. Maximal 16 UTF-8 characters.
hierarchy2	string	Description hierarchy level 2, free format text. Maximal 16 UTF-8 characters.
addId	string	Additional ID/User Login Id, either “Additonalld”/user pin for the subscription process or “User Login Id” for the DECT phone login process with unbound devices. Also used as user credential account for the server authentication process to user configuration file on the server (when <i>login=Id</i> in <i>PPLginVariant</i>).
pin	string	PIN number, for the DECT phone login process with unbound devices - encrypted with public key
sipAuthId	string	SIP authentication user
sipPw	string	SIP authentication password, encrypted with public key
sosNum	string	SOS number
voiceboxNum	string	Voicebox number
manDownNum	string	MANDOWN number
forwardState	CallForwardStateType	Call forwarding setting
forwardTime	integer	Time for call forward
forwardDest	string	Call forward destination
lang	PPLanguageType	User language, read only
holdRingBackTime	integer	Hold ring back time, 0 for off, time steps 1..5
autoAnswer	string	Auto-answer setting for incoming calls. One of “On”, “Off” or “Global”. For more information see also incoming call settings for SIP in 4.34.34 - 4.34.36

microphoneMute	string	Microphone mute setting for incoming calls. One of "On", "Off" or "Global". For more information see also incoming call settings for SIP in 4.34.34 - 4.34.36
warningTone	string	Warning tone setting for incoming calls. One of "On", "Off" or "Global". For more information see also incoming call settings for SIP in 4.34.34 - 4.34.36
allowBargeIn	string	Allow barge in setting for incoming calls. One of "On", "Off" or "Global". For more information see also incoming call settings for SIP in 4.34.34 - 4.34.36
callWaitingDisabled	boolean	"1" or "true", if call waiting is disabled. This value is read only.
external	boolean	"1" or "true", if this data set comes from an external server
trackingActive	boolean	"1" or "true", if the location of this user is being tracked
locatable	boolean	"1" or "true", if the position of this user may be locatable and track able
BTlocatable	boolean	"1" or "true", if the position of this user may be locatable and trackable by a Bluetooth beacon.
BTsensitivity	string	Bluetooth module sensitivity. One of "low", "medium" or "high".
locRight	boolean	"1" or "true", if this user may locate other users which are locatable
msgRight	boolean	"1" or "true", if this user may send messages. Default setting is "1"
sendVcardRight	boolean	"1" or "true", if this user has the permission to send Vcard entries. Default setting is "1"
recvVcardRight	boolean	"1" or "true", if this user has the permission to receive Vcard entries
keepLocalPB	boolean	"1" or "true", if this user may keep the phonebook in the DECT phone after a user logout.
vip	boolean	"1" or "true", if this user is prioritized with its SIP registration.
sipRegisterCheck	boolean	"1" or "true", if this user is used for the PBX SIP register check. Note: Only one user in the system can be set for this register check. Setting this attribute to "false" results to the default setting (the user with the lowest <i>num</i> will automatically set this attribute to true). Setting this attribute to "true" results to setting this attribute to "false" for the user having this attribute set.

allowVideoStream	boolean	"1" or "true", if this user is allowed to use video streaming of video devices.
conferenceServerType	string	Type of the user specific conference server. One of "None", "External", "Integrated", "ExternalBlindTransfer" or "Global".
conferenceServerURI	string	URI for the user specific conference server.
monitoringMode	string	User monitoring state one of "Off", "Passive" or "Aktive".
CUS	MonitoringStateType	Current combined user monitoring state (CUS) of all User monitoring states. Cannot be set.
HAS	MonitoringStateType	Current user monitoring state of the DECT phone assignment status (HAS). Cannot be set.
HSS	MonitoringStateType	Current user monitoring state of the DECT phone subscription status (HSS). Cannot be set.
HRS	MonitoringStateType	Current user monitoring state of the DECT phone registration status (HRS). Cannot be set.
HCS	MonitoringStateType	Current user monitoring state of the DECT phone activity status (HCS). Cannot be set.
SRS	MonitoringStateType	Current user monitoring state of the SIP user registration status (SRS). Cannot be set.
SCS	MonitoringStateType	Current user monitoring state of the silent charging status (SCS). Cannot be set.
CDS	MonitoringStateType	Current user monitoring state of the Call delivery status (CDS). Cannot be set.
HBS	MonitoringStateType	Current user monitoring state of the DECT phone DECT phone battery status (HBS). Cannot be set.
BTS	MonitoringStateType	Current user monitoring state of the DECT phone Bluetooth status (BTS). Cannot be set.
SWS	MonitoringStateType	Current user monitoring state of the DECT phone software status (SWS). Cannot be set.
credentialPw	string	User credential password for user authentication to get the user configuration file from the external server - encrypted with public key. This value is used when the access to the user data server requires user credentials, see <i>Set/Get/EventUserDataServer(Cnf)</i> . This value is normally set at DECT phone during login.
configurationDataLoaded	boolean	"1" or "true", if DECT phone user configuration data (DECT phone profile or user specific configuration data) for this user has been downloaded successfully to the DECT phone. This value is read only.
ppData	string	Configuration data to be downloaded to the DECT phone for this user. The data consists of a sequence of lines containing one setting per line. E. g.: # display-einstellungen UD_Displanguage=en

		UD_DispFont=large UD_DispColor=black # ringer-einstellungen UD_RingerVolumelIntern=level-1 Maximal size is 4 kByte. Note: Before loading the DECT phone profile data into the DECT phone the DECT phone configuration data are merged. First the DECT phone default configuration data (see 4.67.1) are used second the DECT phone profile data (see 4.67.2) of this id are used and finally the user individual DECT phone configuration data.
ppProfileId	integer	Id of the DECT phone profile (see 4.67.1) to be loaded into the DECT phone for this user. If this value is 0, no profile has to be loaded into the DECT phone. Note: Before loading the DECT phone profile data into the DECT phone the DECT phone configuration data are merged. First the DECT phone default configuration data (see 4.67.1) are used second the DECT phone profile data (see 4.67.2) of this id are used and finally the user individual DECT phone configuration data.
fixedSipPort	integer	Explicitly used SIP port for registering the user at the call server. If <i>fixedSipPort</i> =0, no specific port is used for this user and the <i>calculatedSipPort</i> is used.
calculatedSipPort	integer	Calculated SIP port for registering the user at the call server. If <i>calculatedSipPort</i> =0, the <i>fixedSipPort</i> is used. This value is read only.

Which fields are mandatory depend on OMM type (refer to chapter 5), request and response type and permissions of the user being logged in.

CallForwardStateType is an enumeration value defined as follows:

Value	Description
Off	No call forward
Busy	Call forward on busy
NoAnswer	Call forward on no answer
BusyNoAnswer	Call forward on busy and no answer
All	Forward immediately

PPLanguageType is an enumeration value which can have one of these values:

English, German, French, Spanish, Italian, Dutch, Swedish, Portuguese, Danish, Finnish, Norwegian, Czech, Slovakian, Hungarian, [Russian](#), [Turkish](#), Polish, [Estonian](#)

MonitoringStateType is an enumeration value defined as follows:

Value	Description
Unknown	The user monitoring state of this item is unknown
Available	The user monitoring state of this item is available
Warning	The user monitoring state of this item is warning

Unavailable	The user monitoring state of this item is unavailable
Escalated	The user monitoring state of this item is escalated

4.39.2 GetPPUserSummary

This request is sent from the client to the OMM.

With this request the client can query common information about the DECT phone users configured.

Following fields are defined for the request *GetPPUserSummary* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetPPSummary* is an object called *GetPPUserSummaryResp*. It contains following fields in addition to the common attributes:

Name	Type	Description
nRecords	integer	The total number of DECT phones set up.
uidFirst	integer	The User ID of the first DECT phone user set up.
nSipRegistration	integer	The total number of SIP registered DECT phone users
nLocatable	integer	The total number of locatable DECT phone users
nBtLocatable	integer	The total number of Bluetooth locatable DECT phone users
nMsgSend	integer	The total number of locatable DECT phone users allowed messages to send
usersActiveMonitored	integer	The total number of active monitored users
usersPassiveMonitored	integer	The total number of passive monitored users
usersWarned	integer	The total number of users which are in user monitoring <i>warning</i> state
usersUnavailable	integer	The total number of users which are in user monitoring <i>unavailable</i> state
usersEscalated	integer	The total number of users which are in user monitoring <i>escalated</i> state

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.39.3 GetPPUser

With this request the client can acquire one or more DECT phone user data sets from the OMM. The request contains a User ID (*uid*) from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

User IDs usually start at 1, but possibly not all numbers are assigned, i.e. the numbering may not be contiguous.

The responses are ordered by User ID, starting with the smallest.

If the client asked for a certain number records and the given *uid* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given *uid* has to match to a DECT phone user record exactly. If there is a DECT phone user with this *uid*, this record is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetPPUser* (*uid* = "1", *maxRecords* = "20"). If the response contains 20 records, take the last PPN plus one and send another *GetPPUser*. Repeat this until you get less than 20 records back.

Following fields are defined for the request *GetPPUser* in addition to the common attributes:

Name	Type	Mandatory	Description
uid	integer	yes	First User ID of DECT phone users to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to this request is an object called *GetPPUserResp*. It contains following element in addition to the common attributes:

Name	Type	Description
user	sequence of PPUserType	DECT phone user records, if found

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.39.4 CreatePPUser

The client can send this request to OM AXI to create a new DECT phone user data set. Refer to chapter 4.39.1 for information about which attributes of a DECT phone user have to be set in a create request. Consider the license restrictions for this command (see 5.5).

If there is no User ID (*uid*) assigned by the client, the OMM will choose one. If the client assigns a *uid*, it must not exist yet.

Following element is defined for the request *CreatePPUser* in addition to the common attributes:

Name	Type	Mandatory	Description
user	PPUserType	yes	Data of DECT phone user to create.

Depending from the OMM type it may be allowed to create a DECT phone user without any predefined attribute. In this case the client may send an empty *CreatePPUser* request.

If no *pin* is given or if it is empty, it is set to "0000" automatically.

The relation type will be set to *Unbound* automatically.

If the creation was successful, the reply contains all attributes which *GetPPUser* would return. Some data fields may be changed or set by the OMM. If the creation fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreatePPUserResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.39.5 CreateFixedPP

With this request the client can create a DECT phone user and Device with a *Fixed* relation. Refer to chapter 4.38.1 and 4.39.1 for information about which attributes of a DECT phone user and Device have to be set in a create request.

Consider the license restrictions for this command (see 5.5).

If there is no User ID (*uid*) or PPN assigned by the client, the OMM will choose one. If the client assigns a *uid* and/or *ppn*, it must not exist yet.

The request will set up two basic data sets atomically: A DECT phone device data set and a DECT phone user data set.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
user	PPUserType	yes	Data of DECT phone user to create.
pp	PPDevType	yes	Data of DECT phone to create.

The relation type will be set to Fixed automatically.

If the creation was successful, the reply contains all attributes which *GetPPUser* and *GetPPDevice* would return. Some data fields may be changed or set by the OMM. If the creation fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

If no *pin* is given or if it is empty, it is set to "0000" automatically.

The reply to this request is called *CreateFixedPPResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.39.6 DeletePPUser

With this request the client can delete a DECT phone user record in the OMM. The key is the User Id (*uid*).

Following fields are defined for the request *DeletePPUser* in addition to the common attributes:

Name	Type	Mandatory	Description
uid	integer	no	User ID of DECT phone user to be deleted
num	string	no	Phone number or SIP user ID.

If the DECT phone user has a dynamic relation to a DECT phone device they are unlinked automatically. If they have a fixed relation, User and Device will be deleted.

The reply to this request is an object called *DeletePPUserResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.39.7 SetPPUser

The client can send this request to OM AXI to change a DECT phone user data set. The User ID (*uid*) has to be filled in by the client to identify the record to be changed. Additionally the attributes which have to be changed must be filled in by the client. Refer to chapter 4.39.1 for information about which attributes of a DECT phone user may be changed using *SetPPUser*. Consider the license restrictions for this command (see 5.5).

Following element is defined for the request *SetPPUser* in addition to the common attributes:

Name	Type	Mandatory	Description
user	PPUserType	yes	Data of DECT phone user to change.

If the change was successful, the reply contains all attributes which *GetPPUser* would return. Some data fields may be changed or set by the OMM. If the change fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

If a *pin* is given and its value is empty, it is set to "0000" automatically.

It is not possible to change a relationship to a DECT phone device using this request. The attributes *relType* and *ppn* are ignored.

The reply to this request is called *SetPPUserResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

Value	Description
EForbidden	It is not allowed to change DECT phone user data, if the user data set is located on an external user data server.
ELicense	It is not allowed to locate or track this DECT phone user, because of license setting

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.39.8 EventPPUserCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *PPUserCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with only a *uid* in the *user* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
user	PPUserType	yes	Changed attributes
deleted	boolean, default false		„1” or “true”, if this data set has been deleted

Available filters: uid

4.39.9 EventPPUserSummary

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf* or to *PPUserSummary*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes.

Attributes sent in this notification:

Name	Type	Description
nRecords	integer	The total number of DECT phones set up.
uidFirst	integer	The User ID of the first DECT phone user set up.
nSipRegistration	integer	The total number of SIP registrationed DECT phone users
nLocatable	integer	The total number of locatable DECT phone users
nBtLocatable	integer	The total number of Bluetooth locatable DECT phone users
nMsgSend	integer	The total number of DECT phone users allowed to send messages
usersActiveMonitored	integer	The total number of active monitored users
usersPassiveMonitored	integer	The total number of passive monitored users
usersWarned	integer	The total number of users which are in user monitoring <i>warning</i> state
usersUnavailable	integer	The total number of users which are in user monitoring <i>unavailable</i> state
usersEscalated	integer	The total number of users which are in user monitoring <i>escalated</i> state

4.40 User Device Relation Configuration And Maintenance

4.40.1 SetPPUserDevRelation

The client can send this request to OM AXI to change the relation between a DECT phone user and a DECT phone device data set. The User ID (*uid*) has to be filled in by the client to identify the DECT phone user data set to be changed. Additionally the relation attribute must be filled in by the client.

Following element is defined for the request *SetPPUserDevRelation* in addition to the common attributes:

Name	Type	Mandatory	Description
uid	integer	yes	User ID, of the user data set.
relType	PPRelTypeType	yes	Type or state of a relationship to the DECT phone device to be set.

It is only possible to change a relationship between a user and a DECT phone device from *Fixed* to *Dynamic* and from *Dynamic* to *Fixed*.

The reply to this request is called *SetPPUserDevRelationResp*. It contains all actual set attributes, all attributes are optional. The Event notification for this change is done by *EventPPDevCnf* and *EventPPUserCnf*.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.41 Telephony Monitoring

4.41.1 GetPPState

With this request the client can query information about the current DECT phone state.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
ppn	integer	yes	Portable part number being requested

The reply to this request is an object called *GetPPStateResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
ppn	integer	yes	Portable part number being requested.
onHook	boolean	yes	„1” or “true”, if this DECT phone is in onHook state.
silentCharging	boolean	yes	„1” or “true”, if this DECT phone is in silent charging state.
callState	PPUserCallStateType	yes	Current call state info of the user of this DECT phone
batteryLevel	integer	no	Current percentage value (0 – 100%) of the DECT phones battery state. Missing, when the DECT phones battery state is unknown or cannot be detected. Note: The value will only be sent when user monitoring is activated for the user of this DECT phone device.
bluetooth	boolean	no	„1” or “true”, if this DECT phone is in Bluetooth operational mode.
swVersion	string	no	Current DECT phones software version.
registered	boolean	no	„1” or “true”, if this DECT phone has successfully preformed a SIP registration.. Only contained, if a user is assigned to this DECT phone.
regServerType	string	no	Used SIP registrar server the user is registered. One of “None”, “Primary”, “Secondary” or “Tertiary”. Only contained, if a user is assigned to this DECT phone.
regServerAddr	string	no	Used SIP registrar server address the user is registered. Only contained, if a user is assigned to this DECT phone.

regServerPort	integer	no	Used SIP registrar server port the user is registered. Only contained, if a user is assigned to this DECT phone.
---------------	---------	----	--

Possible values for *errorCode*:

Value	Description
ENoEnt	The DECT phone was not available to deliver its state. It was not in a working state (e. g. IPEI is missing or no user is logged in).
EFailed	The DECT phone currently cannot deliver its state.

For further possible *errorCode* values see 3.3.

The client needs one of these permissions to use this request: *Monitoring*

4.41.2 EventPPState

This event is sent by OM AXI telephony state of a DECT phone has changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to one of *PPState*, *PPDevState*, *PPCallState*, *PPBtState*, *PPSipState*, *PPCallStateCalling*, *PPCallStatePaging*, *PPCallStateConnected*, *PPCallStateIdle*, *PPCallStateNone*, *PPDevStateOnHook*, *PPDevStateSilentCharging*, *PPDevStateBatteryLevel*, *PPDevStateSwVersion*, *PPSipStateRegEvent*, *PPSipStateEndEvent*, *PPSipStateNotifyEvent*, *PPSipStateRegistered*, *PPSipStateServerType*, *PPSipStateServerAddr* or *PPSipStateServerPort*. This is allowed if the client has at least one of the following permissions: *Monitoring*.

When a client wants to stay updated about the telephony state of a DECT phone, it can subscribe to notifications of type *PPCallState* for one or more DECT phones.

Each time one of these DECT phones changes its state (depending on the current subscription and DECT phone state change), an event of this type is sent by OM AXI.

This event contains following fields:

Name	Type	Mandatory	EventType	Description
ppn	integer	yes		PPN being monitored
onHook	boolean	no	<i>PPState</i> <i>PPDevState</i> <i>PPDevStateOnHook</i>	„1” or “true”, if this DECT phone is in onHook state
silentCharging	boolean	no	<i>PPState</i> <i>PPDevState</i> <i>PPDevStateSilentCharging</i>	„1” or “true”, if this DECT phone is in silent charging state
batteryLevel	integer	no	<i>PPState</i> <i>PPDevState</i> <i>PPDevStateBatteryLevel</i>	Current percentage value (0 – 100%) of the DECT phones battery state. Note: The value will only be sent when user monitoring is activated for the user of this DECT

				phone device.
swVersion	string	no	<i>PPState</i> <i>PPDevState</i> <i>PPDevStateSwVersion</i>	Current DECT phones software version.
callState	PPUserCallStateType	yes	<i>PPState</i> <i>PPCallState</i> <i>PPCallStateCalling</i> <i>PPCallStatePaging</i> <i>PPCallStateConnected</i> <i>PPCallStateIdle</i> <i>PPCallStateNone</i> <i>PPDevStateOnHook</i>	Current call state info of the user of this DECT phone
bluetooth	boolean	no	<i>PPState</i> <i>PPBtState</i>	„1” or “true”, if this DECT phone is in Bluetooth operational mode.
sipEvent	PPUserSIPEventType	no	<i>PPState</i> <i>PPSipState</i> <i>PPSipStateRegEvent</i> <i>PPSipStateEndEvent</i> <i>PPSipStateNotifyEvent</i>	New SIP event info for the user of this DECT phone
registered	boolean	no	<i>PPState</i> <i>PPSipState</i> <i>PPSipStateRegistered</i>	„1” or “true”, if this DECT phone has successfully preformed a SIP registration..
regServerType	string	no	<i>PPState</i> <i>PPSipState</i> <i>PPSipStateServerType</i>	Used SIP registrar server the user is registered. One of “None”, “Primary”, “Secondary” or “Tertiary”.
regServerAddr	string	no	<i>PPState</i> <i>PPSipState</i> <i>PPSipStateServerAddr</i>	Used SIP registrar server address the user is registered.
regServerPort	integer	no	<i>PPState</i> <i>PPSipState</i> <i>PPSipStateServerPort</i>	Used SIP registrar server port the user is registered.

Available filters: *ppn*

4.42 RFP Configuration And Maintenance

4.42.1 RFPTType

This type contains all configuration data fields and state information about an RFP. It is used in different requests and responses defined in this document. Not all fields are used in all OMM versions and in all Request and Response types.

Name	Type	Description
id	integer	Unique RFP identifier. The numbering starts at 0
ethAddr	string	Ethernet address, format "00:11:22:aa:bb:cc"
dectOn	boolean	„1” or “true”, if DECT is enabled on this RFP
wlanOn	boolean	„1” or “true”, if WLAN is enabled on this RFP
licenseRfp	boolean	This RFP is used for licensing, it cannot be deleted
name	string	Name of this RFP, free format text.
hierarchy1	string	Position hierarchy level 1, free format text
hierarchy2	string	Position hierarchy level 2, free format text
hierarchy3	string	Position hierarchy level 3, free format text
hierarchy4	string	Position hierarchy level 4, free format text
rpn	integer	DECT RPN
pagingArea	integer	Paging area number, -1 for unassigned
cluster	integer	Synchronization cluster, 0 is invalid
preferredSync	boolean	This RFP is preferred in sync start up
reflectiveEnv	boolean	This RFP is located in a reflective environment
site	integer	Reference to site data set id.
x	integer	X coordinate for visualization
y	integer	Y coordinate for visualization
hwType	RFPHwTypeType	Type of hardware for this RFP
hwTypeLocked	boolean	Read only value to indicate the possibility to configure the <i>hwType</i> . Until the RFP did not connect ever to the OMM the value is false and <i>hwType</i> can be configured. The value never changes from true to false!
wlanProfile	integer	WLAN profile 0 is invalid
wlanAntennaDiv	boolean	„1” or “true”, if WLAN antenna diversity is set

wlanAntenna ¹	integer	Selected WLAN antenna, 1 or 2.
wlanChannel	integer	Configured WLAN channel, one of 1 to 14, 36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 147, 151, 155, 159, 163, 167, 171
wlanHighThroughput	boolean	„1” or “true”, if WLAN high throughput mode 40 is set
wlanPower	integer	Cofigured WLAN transmit power in percent, one of 6, 12, 25, 50, 100.
conferenceChannels	boolean	„1” or “true”, if this RFP can be used for conference cahnnels
<i>following attributes are transient (selected with withState in Requests):</i>		
connected	boolean	„1” or “true”, if this RFP is connected, transient
ipAddr	string	Current or last known IP address, transient
newSoftwareRequest	boolean	„1” or “true”, if this RFP requested the OMM to load new software., transient
dectRunning	boolean	„1” or “true”, if the DECT is running on this RFP., transient
wlanRunning	boolean	„1” or “true”, if the WLAN is running on this RFP, transient
ommRunning	boolean	„1” or “true”, if the OMM is running on this RFP, transient
ommStbRunning	boolean	„1” or “true”, if the standby OMM is running on this RFP, transient
hasWlan	boolean	„1” or “true”, if WLAN is supported by this RFP hardware
hasEncryption	boolean	„1” or “true”, if encryption is supported by this RFP hardware These features can be configured within a site
hasAdvancedFeatures	boolean	„1” or “true”, if Hi-Q audio technology (wide band) G.722, terminal video, DECT security and secure SIP are supported by this RFP hardware. These features can be configured within a site.
syncState	RFPSyncStateType	Current DECT synchronization state for this RFP.
swVersion	string	Current software version on this RFP in the format: <majorRelease>.<minorRelease>.{RC x SP y <bugfixVersion>} [Build z] [version description],

¹ Only RFP42 Type

		e. g. „2.1 RC4“, „2.1 SP1“, „2.1.5“, „2.1SP1 Build 2“,r „3.0 RC1 Build 1 - OpenMobility SIP 3.0RC1-Internal“. RCx: Release candidate number x SPy: Release service pack number y bugfixVersion: Old bugfix release version number Build z: Build release number z for internal use
brandingMismatch	boolean	„1” or “true”, if branding of this RFP and OMM do not fit together
versionMismatch	boolean	„1” or “true”, if software version of this RFP and OMM do not fit together
stbMismatch	boolean	„1” or “true”, if this RFP has an invalid OMM standby configuration.
wlanLinkNok	boolean	„1” or “true”, if the Ethernet link is too slow (e. g. only 10 MBit/s) to enable WLAN on this RFP
wlanChannelUsed	integer	Used Channel one of 1 to 14, 36, 40, 44, 48, 52, 56, 60, 64, 100, 104, 108, 112, 116, 120, 124, 128, 132, 136, 140, 147, 151, 155, 159, 163, 167, 171. This value is read only.
wlanHighThroughputTypeUsed	string	Used high throughput channel type, one of “None”, “HT20”, “HT40Minus” or “HT40Plus” . This value is read only.
wlanPowerUsed	integer	Used WLAN transmit power in DBM (decibel per milliwatt) . This value is read only.
nSyncRels	integer	Number of DECT synchronization relations
<i>following attributes are detailed (selected with withDetails in Requests):</i>		
radioType	RFPRadioTypeType	The DECT radio type of this RFP.
outdoorType	boolean	„1” or “true”, if this is an outdoor RFP hardware
hasFreqShift	boolean	„1” or “true”, if frequency shift is supported for this RFP

In a *CreateRFP* request the fields *name* and *ethAddr* are mandatory. In a *SetRFP* request the field's *id* is mandatory. In all other case the fields are optional. The success of the requests depend further on the permissions of the user being logged in.

The transient data field can only be known to the OMM when the RFP is connected or has been connected before.

These are possible values for *RFPHwTypeType*:

Value	Description
Unknown	Hardware type not known
RFP 31	Non WLAN indoor RFP
RFP 33	Non WLAN outdoor RFP

RFP 41	WLAN indoor RFP
RFP 42	WLAN specific DECT (e. g. with encryption) indoor RFP
RFP 42 US	WLAN specific DECT (e. g. with encryption) indoor RFP / US variant
RFP 32	Non WLAN specific DECT (e. g. with encryption) indoor RFP
RFP 32 US	Non WLAN specific DECT (e. g. with encryption) indoor RFP / US variant
RFP 34	Non WLAN specific DECT (e. g. with encryption) outdoor RFP
RFP 34 US	Non WLAN specific DECT (e. g. with encryption) outdoor RFP / US variant
RFP 35	Non WLAN indoor NG RFP
RFP 36	Non WLAN outdoor NG RFP
RFP 37	Non WLAN outdoor NG RFP with outside DECT antennas
RFP 43	WLAN indoor NG RFP
RFP L33	Non WLAN outdoor licensed RFP
RFP L41	WLAN indoor licensed RFP
RFP L42	WLAN specific DECT (e. g. with encryption) indoor licensed RFP
RFP L42 US	WLAN specific DECT (e. g. with encryption) indoor licensed RFP / US variant
RFP L32	Non WLAN specific DECT (e. g. with encryption) indoor licensed RFP
RFP L32 US	Non WLAN specific DECT (e. g. with encryption) indoor licensed RFP / US variant
RFP L34	Non WLAN specific DECT (e. g. with encryption) outdoor licensed RFP
RFP L34 US	Non WLAN specific DECT (e. g. with encryption) outdoor licensed RFP / US variant
RFP L35	Non WLAN indoor NG licensed RFP
RFP L36	Non WLAN outdoor NG licensed RFP
RFP L37	Non WLAN outdoor NG licensed RFP with outside DECT antennas
RFP L43	WLAN indoor NG licensed RFP
RFP SL36	Non WLAN outdoor single cell NG RFP
RFP SL37	Non WLAN outdoor NG single cell RFP with outside DECT antennas
RFP SL43	WLAN indoor NG single cell RFP
PC ECM	External (linux PC) conference mixer that is handled as an RFP in the OMM

Note: Not all RFP HW types are currently existing or in use!

These are the possible values for RFPRadioTypeType:

Value	Description
LowTX	North American TX power limit
NormalTX	European TX power limit
ConfigurableTX	The power limit can be configured
None	No power limit for this RFP

These are the possible values for RFPSyncStateType:

Value	Description
Inactive	RFP not DECT active
NotSynced	RFP is not synchronized
Searching	RFP searches for other synchronized RFPs
Synced	RFP is synchronized

4.42.2 GetRFPSummary

This request is sent from the client to the OMM.

With this request the client can query common information about the RFPs configured.

Following fields are defined for the request *GetRFPSummary* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetRFPSummary* is an object called *GetRFPSummaryResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
nRFPs	integer	yes	The total number of RFPs set up.
idFirst	integer	yes	The id of the first RFP set up.
nConnected	integer	yes	The total number of connected RFPs.
wrongBrandedRFPs	integer	yes	The total number of RFPs detecting a branding mismatch
wrongStandbyRFPs	integer	yes	The total number of RFPs detecting an OMM standby misconfiguration
wrongVersionedRFPs	integer	yes	The total number of RFPs having a SW version mismatch

newAvailSWRFPs	integer	yes	The total number of RFPs detecting a new SW version to load
usbOverloads	integer	yes	The total number of RFPs detecting usb overload capacity
advancedFeaturesErrorRFPs	integer	yes	The total number RFPs detecting an advanced feature error (Hi-Q audio technology (wide band) G.722, terminal video, DECT security and secure SIP).
DecryptedDECTRFPs	integer	yes	The total number of RFPs not supporting encryption.
DECTactivatedRFPs	integer	yes	The total number of DECT activated RFPs
DECTactiveRFPs	integer	yes	The total number of DECT active RFPs
WLANactivatedRFPs	integer	yes	The total number of WLAN activated RFPs
WLANrunningRFPs	integer	yes	The total number of WLAN running RFPs
usedDECTclusters	integer	yes	The total number of used DECT clusters
usedWLANprofiles	integer	yes	The total number of used WLAN profiles
usedPagingAreas	integer	yes	The total number of used paging areas.

For possible `errCode` values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.42.3 GetRFP

With this request the client can query information about one or more RFPs from the OMM.

The request contains an RFP-ID from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

RFP-IDs usually start at 0, but possibly not all numbers are assigned, i.e. the data set numbering may not be contiguous.

Only attributes of the RFP used on the present OMM version and accessible with the current user's permissions are returned. Refer to chapter 4.48 for details.

The responses are ordered by *id*, starting with the smallest.

If the client asked for a certain number records but the given *id* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given *id* has to match to a record exactly. If there is a record with this *id*, it is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetRFP* (*id* = "0", *maxRecords* = "20"). If the response contains 20 records, take the last *id* plus one and send another *GetRFP*. Repeat this until you get less than 20 records back.

Following fields are defined for the request *GetRFP* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	First RFP id of RFPs to get

maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.
withState	boolean, default false	no	„1” or “true”, if state information is also wanted
withDetails	boolean, default false	no	„1” or “true”, if details are also wanted.

The reply to *GetRFP* is an object called *GetRFPResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
rfp	sequence of RFPTYPE	no	RFP record, if found

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.42.4 CreateRFP

The client can send this request to OM AXI to create a new RFP data set. Refer to chapter 4.42.1 for information about which attributes of an RFP have to be set in a create request. Consider the license restrictions for this command (see 5.5).

If there is no *id* assigned by the client, the OMM will choose one. If the client assigns an *id*, it must not exist yet.

Note that attributes describing transient properties are ignored, e. g. *connected*.

Following fields are defined for the request *CreateRFP* in addition to the common attributes:

Name	Type	Mandatory	Description
rfp	RFPTYPE	yes	Data of RFP to create.

If the RFP creation was successful, the reply contains all RFP attributes which *GetRFP* would return, but no state information. Some data fields may be changed or set by the OMM. If the creation fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreateRFPResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

Value	Description
EFailed	RFP could not be created, e. g. because the DECT regulatory domain has not been set before
ENoMem	No more RFPs can be created - the system is completely configured
ELicense	No more RFPs can be created because of license limitation

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.42.5 DeleteRFP

With this request the client can delete an RFP record in the OMM. The key used to delete an RFP is its *id*. Consider the license restrictions for this command (see 5.5).

Following fields are defined for the request *DeleteRFP* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	id of RFP to delete

The reply to *DeleteRFP* is an object called *DeleteRFPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

Possible values for *errCode*:

Value	Description
ELicense	This operation is not permitted with that instance, because of a licensed RFP

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.42.6 SetRFP

The client can send this request to OM AXI to change an RFP data set. The *id* has to be filled in by the client to identify the record to be changed. Additionally the attributes of the RFP which have to be changed must be filled in by the client.

Note that attributes describing transient properties are ignored.

Following fields are defined for the request *SetRFP* in addition to the common attributes:

Name	Type	Mandatory	Description
rfp	RFPTYPE	yes	Data of RFP to change.

If the RFP change was successful, the reply contains all RFP attributes which *GetRFP* would return, but no state information. Some data fields may be changed or set by the OMM. If the creation fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetRFPResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.42.7 EventRFPCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *RFPCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Configuration items are all attributes which are not selected with *withState* or with *withDetails*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

This event is also sent upon creation of an RFP. If the RFP has been deleted, this event is sent with only an *rfpld* in the *rfp* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
rfp	RFPTtype	yes	Contains all attributes which have been changed.
deleted	boolean, default false		„1” or “true”, if this data set has been deleted

Available filters: rfpId

4.42.8 EventRFPState

This event is sent by OM AXI when a change of a state attribute happened. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *RFPState*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Items belonging to this group are those which are selected with *withState*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

This event is not sent upon deletion of an RFP.

Attributes sent in this notification:

Name	Type	Mandatory	Description
id	integer	yes	Id of the RFP
rfp	RFPTtype	yes	Contains all attributes which have been changed.

Available filters: rfpId

4.42.9 EventRFPDetails

This event is sent by OM AXI when RFP details have been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *RFPDetails*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Items belonging to this group are those which are selected with *withDetails*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

This event is not sent upon deletion of an RFP.

Attributes sent in this notification:

Name	Type	Mandatory	Description
id	integer	yes	Id of the RFP
rfp	RFPTtype	yes	Contains all attributes which have been changed.

Available filters: rfpId

4.42.10 EventRFPConnectAttempt

This event is sent by OM AXI when a non-configured RFP tries to connect. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *RFPConnectAttempt*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Items belonging to this group are those which are selected with *withDetails*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

This event is not sent upon deletion of an RFP.

Attributes sent in this notification:

Name	Type	Mandatory	Description
ethAddr	string	yes	Ethernet address of the RFP, format like "00:11:22:33:44:55:66"
ipAddr	string	yes	IP address of the RFP in dot-notation

Available filters: (none)

4.42.11 EventRFPSummary

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf* or to *RFPSummary*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes.

Attributes sent in this notification:

Name	Type	Description
nRFPs	integer	The total number of RFPs set up.
idFirst	integer	The id of the first RFP set up.
nConnected	integer	The total number of connected RFPs.
wrongBrandedRFPs	integer	The total number of RFPs detecting a branding mismatch

wrongStandbyRFPs	integer	The total number of RFPs detecting an OMM standby misconfiguration
wrongVersionedRFPs	integer	The total number of RFPs detecting a SW version mismatch
newAvailSWRFPs	integer	The total number of RFPs detecting a new SW version to load
usbOverloads	integer	The total number of RFPs detecting usb overload capacity
advancedFeaturesErrorRFPs	integer	The total number RFPs detecting an advanced feature error (Hi-Q audio technology (wide band) G.722, terminal video, DECT security and secure SIP).
DecryptedDECTRFPs	integer	The total number of RFPs not supporting encryption.
DECTactivatedRFPs	integer	The total number of DECT activated RFPs
DECTactiveRFPs	integer	The total number of DECT active RFPs
WLANactivatedRFPs	integer	The total number of WLAN activated RFPs
WLANrunningRFPs	integer	The total number of WLAN running RFPs
usedDECTclusters	integer	The total number of used DECT clusters
usedWLANprofiles	integer	The total number of used WLAN profiles
usedPagingAreas	integer	The total number of used paging areas

4.42.12 CapturedRFPTType

This type contains all data fields about a captured RFP. It is used in different requests and responses defined in this document.

Name	Type	Description
ethAddr	string	Ethernet address, format "00:11:22:aa:bb:cc"
ipAddr	string	Current or last known IP address, transient
hwType	RFPHwTypeType	Type of hardware for this RFP
hasWlan	boolean	„1” or “true”, if WLAN is supported by this RFP hardware
radioType	RFPRadioTypeType	The DECT radio type of this RFP.
outdoorType	boolean	„1” or “true”, if this is an outdoor RFP hardware
hasFreqShift	boolean	„1” or “true”, if frequency shift is supported for this RFP

4.42.13 SetRFPCapture

With this request the client can set the RFP capturing. After setting the OMM automatically captures all not configured RFPs connecting the OMM. The client can use these RFPs for an easier configuration.

Following fields are defined for the request *SetRFPCapture* in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, to enable the RFP capturing.

The reply is an object called *SetRFPCaptureResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.42.14 GetRFPCapture

A client can get a list with all captured but not configured RFPs.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(none)

The reply to this request is an object called *GetRFPCaptureResp*. It contains following element in addition to the common attributes:

Name	Type	Description
enable	boolean	„1” or “true”, when RFP capturing is enabled.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.42.15 EventRFPCaptureCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, when RFP capturing is enabled.

Available filters: (none)

4.42.16 GetRFPCaptureList

A client can get a list with all captured but not configured RFPs.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(none)

The reply to this request is an object called *GetRFPCaptureListResp*. It contains following element in addition to the common attributes:

Name	Type	Description
rfp	sequence of capturedRFPTtype	All captured RFP records, if any

Possible values for *errCode*:

Value	Description
ENoEnt	No captured RFP exists

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.42.17 DeleteRFPCaptureList

A client can delete the list with all captured RFPs.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			None

The reply to this request is an object called *DeleteRFPCaptureListResp*. It contains following element in addition to the common attributes:

Name	Type	Description
		(none)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.42.18 DeleteRFPCaptureListElem

A client can delete one captured RFP from the captured RFPs list.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
ethAddr	string	yes	Ethernet address of the license RFP, format "00:11:22:aa:bb:cc"

The reply to this request is an object called *DeleteRFPCaptureListElemResp*. It contains following element in addition to the common attributes:

Name	Type	Description
		(none)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.42.19 EventRFPCapture

This event is sent by OM AXI when a not configured RFP connects the OMM. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the RFP capturing state is set in the OMM and the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
Deleted	boolean	yes	„1” or “true”, when this captured RFP was deleted from the list, otherwise the RFP is added to the list.
Rfp	capturedRFPTType	yes	Data of the captured RFP that changes.

Available filters: (none)

4.43 RFPM Configuration And Maintenance

The RFP management is configuration and Maintenance is only available for SIP-DECT with CloudId.

4.43.1 IPAddressType

This type contains attribute values for an RFP list element.

Name	Type	Description
ipAddr	string	IP address list element.

4.43.2 GetRFPM

With this request the client can query information about RFP management from the OMM.

Following fields are defined for the request *GetRFPM* in addition to the common attributes:

Name	Type	Mandatory	Description

The reply to *GetRFPM* is an object called *GetRFPMResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
useDHCP	boolean	yes	„1” or “true”, if the RFP running the OMM provides a DHCP server
firstIpAddr	string	yes	First IP address the DHCP server of RFP running the OMM provides for DHCP clients
lastIpAddr	string	yes	Last IP address the DHCP server of RFP running the OMM provides for DHCP clients
useLocalCfg	boolean	yes	„1” or “true”, if the OMM RFP management uses a local configuration
ipAddr	string	yes	RFP IP address running the OMM
netmask	string	yes	Netmask to be used for all RFPs
gateway	string	yes	Gateway to be used for all RFPs
dnsServer	sequence of IPAddressType	yes	List of DNS servers to be used for all RFPs
dnsDomain	string	yes	DNS domain name to be used for all RFPs

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.43.3 SetRFP

The client can send this request to OM AXI to change an RFP management data set.

Following fields are defined for the request *SetRFP* in addition to the common attributes:

Name	Type	Mandatory	Description
useDHCP	boolean	no	„1” or “true”, if the RFP running the OMM provides a DHCP server
firstIpAddr	string	no	First IP address the DHCP server of RFP running the OMM provides for DHCP clients
lastIpAddr	string	no	Last IP address the DHCP server of RFP running the OMM provides for DHCP clients
useLocalCfg	boolean	no	„1” or “true”, if the OMM RFP management uses a local configuration
ipAddr	string	no	RFP IP address running the OMM
netmask	string	no	Netmask to be used for all RFPs
gateway	string	no	Gateway to be used for all RFPs

dnsServer	sequence of IPAddressType	no	List of DNS servers to be used for all RFPs
dnsDomain	string	no	DNS domain name to be used for all RFPs

If the RFP management data change was successful, the reply contains all RFPM attributes which *GetRFPMResp* would return.

The reply to this request is called *SetRFPMResp*. It contains all actual set attributes.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.43.4 EventRFPMConf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification are the same to *GetRFPMResp*.

Available filters: None

4.44 WLAN Regulatory Domain Configuration And Maintenance

4.44.1 GetWLANRegDomainList

With this request the client can query all WLAN regulatory domains the possible channels.

4.44.1.1 WLANRegDomainType

This type contains attribute values for a WLAN regulatory domain.

Name	Type	Description
name	string	WLAN regulatory domain. The possible values are a 2 character string as country code for the regulatory domain. Valid strings are defined in ISO-3166-1.

Following fields are defined for the request *GetWLANRegDomainList* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetWLANRegDomainListResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
regDomain	sequence of WLANRegDomainType	yes	All WLAN regulatory domains.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.44.2 GetWLANRegDomain

With this request the client can query the WLAN regulatory domain.

Following fields are defined for the request *GetWLANRegDomain* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetWLANRegDomainResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
regDomain	string	yes	WLAN regulatory domain. The possible values are a 2 character string as country code for the regulatory domain. Valid strings are defined in ISO-3166-1.
channels2_4G	string	yes	Comma separated list of possible 2.4GHz channels for this WLAN regulatory domain.
Channels5G	string	yes	Comma separated list of possible 5GHz channels for this WLAN regulatory domain.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.44.3 SetWLANRegDomain

With this request the client can set the WLAN regulatory domain.

Following fields are defined for the request *SetWLANRegDomain* in addition to the common attributes:

Name	Type	Mandatory	Description
regDomain	string	yes	WLAN regulatory domain. The possible values are a 2 character string as country code for the regulatory domain. Valid strings are defined in ISO-3166-1.

The reply is an object called *SetWLANRegDomainResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
regDomain	string	yes	WLAN regulatory domain. The possible values are a 2 character string as country code for the regulatory domain. Valid strings are defined in ISO-3166-1.
channels2_4G	string	yes	Comma separated list of possible 2.4GHz channels for this WLAN regulatory domain.
Channels5G	string	yes	Comma separated list of possible 5GHz channels for this WLAN regulatory domain.

For possible `errCode` values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.44.4 EventWLANRegDomainCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
regDomain	string	yes	Changed WLAN regulatory domain. The possible values are a 2 character string as country code for the regulatory domain. Valid strings are defined in ISO-3166-1.
channels2_4G	string	yes	Comma separated list of possible 2.4GHz channels for this WLAN regulatory domain.
Channels5G	string	yes	Comma separated list of possible 5GHz channels for this WLAN regulatory domain.

Available filters: (none)

4.45 WLAN Profile Configuration And Maintenance

4.45.1 SSIDProfileType

This type contains all configuration data fields for an SSID profile.

Name	Type	Description
ssid	string	SSID
enable	boolean	„1” or “true”, to enable this SSID profile
vlanTag	integer	VLAN tag for this SSID, 0 for none.
security	WLANSecurityType	One of "Open", "WEP", "WPA"
bssIsolation	boolean	„1” or “true”, to enable BSS isolation
useACL	boolean	„1” or “true”, to use the ACL (MAC address filter)
hiddenSSID	boolean	„1” or “true”, to hide the SSID
<i>Settings relevant for WEP:</i>		
privacy	boolean	„1” or “true”, to enable shared authentication
wepKey1	string	WEP key 1

wepKey2	string	WEP key 2
wepKey3	string	WEP key 3
wepKey4	string	WEP key 4
defaultWepKey	integer	Default WEP key to be used, 1..4
wepKeyHex	boolean	„1” or “true”, if the WEP keys are in hexadecimal form, otherwise they are text
keyLength	integer	WEP key length. One of 64, 128, 256
<i>Settings relevant for WPA:</i>		
wpaType	WpaTypeType	One of "Any", "WPA1", "WPA2"
psk	string	PSK for WPA
pskHex	boolean	„1” or “true”, if the WEP keys are in hexadecimal form, otherwise they are text
useRadius	boolean	„1” or “true”, to use a Radius server instead of PSK
distributionIntval	integer	Key distribution interval in seconds
<i>Additional WPA settings relevant for Radius:</i>		
radiusAddr	string	IP address of radius server
radiusPort	integer	Port number of radius server
radiusSecret	string	Password of radius server

If WEP is used, at least one WEP key has to be filled in; the list must always start with Key 1 and must be contiguous.

The security type *"Radius"* implies WEP encryption. This must not be confused with combining WPA with Radius which is enabled by security type *"WPA"* and *useRadius*.

4.45.2 WLANProfileType

This type contains all configuration data fields for a WLAN profile.

Name	Type	Description
id	integer	ID of this profile, 0 is invalid
enable	boolean	„1” or “true”, to enable this profile
nRFPs	integer	Number of RFPs actual assigned to this profile. This value is read only.
beaconIntval	integer	Beacon interval

dtimIntval	integer	DTIM beacon interval
rtsThreshold	integer	RTS Threshold in bytes
fragThreshold	integer	Fragmentation Threshold in bytes
mode	WLANModeType	One of "B", "G", "BG", "ABG" (only RFP43), "N" (only RFP43)
qos	QoSType	One of "None", "WME", "WME-VLAN" or "WME-DiffServ". Valid values for RFP42: <i>None</i> , <i>WME-VLAN</i> or <i>WME-DiffServ</i> Valid values for RFP43: <i>None</i> or <i>WME</i>
profileType	string	The profile type based on the RFP type. One of: "RFP42" or "RFP43".
ssid1	SSIDProfileType	First SSID element
ssid2	SSIDProfileType	Second SSID element
ssid3	SSIDProfileType	Third SSID element
ssid4	SSIDProfileType	Forth SSID element
interferenceAvoidance	boolean	„1" or "true", to enable protection against interferences. Used only for RFP42 type.
maxRate	integer	Maximal data rate in kbps, one of: 1000, 2000, 5500, 6000, 9000, 11000, 12000, 18000, 22000, 24000, 36000, 48000, 54000. Used only for RFP42 type.

4.45.3 GetWLANProfile

With this request the client can acquire one or more WLAN profile data sets from the OMM. The request contains an ID (*id*) from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

User IDs usually start at 1, but possibly not all numbers are assigned, i.e. the numbering may not be contiguous.

The responses are ordered by *id*, starting with the smallest.

If the client asked for a certain number records and the given *id* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given *id* has to match to a record exactly. If there is a record with this *id*, this record is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetWLANProfile* (*id* = "1", *maxRecords* = "20"). If the response contains 20 records, take the last *id* plus one and send another *GetWLANProfile*. Repeat this until you get less than 20 records back.

Following fields are defined for the request *GetWLANProfile* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	First ID of WLAN profiles to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to this request is an object called *GetWLANProfileResp*. It contains following element in addition to the common attributes:

Name	Type	Description
profile	sequence of WLANProfileType	WLAN profile records, if found

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.45.4 CreateWLANProfile

The client can send this request to OM AXI to create a new WLAN profile.

If there is no *id* assigned by the client (or it is 0), the OMM will choose one. If the client assigns an *id*, it must not exist yet.

Following element is defined for the request *CreateWLANProfile* in addition to the common attributes:

Name	Type	Mandatory	Description
profile	WLANProfileType	yes	Data of WLAN profile to create.

If the creation was successful, the reply contains all attributes which *GetWLANProfile* would return. Some data fields may be changed or set by the OMM. If the creation fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreateWLANProfileResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.45.5 DeleteWLANProfile

With this request the client can delete a WLAN profile in the OMM. The key is the *id*.

Following fields are defined for the request *DeleteWLANProfile* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	ID of the WLAN profile to be deleted

The reply to this request is an object called *DeleteWLANProfile*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

Possible values for *errCode*:

Value	Description
EFailed	Profile cannot be deleted because there are still RFPs in it

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.45.6 SetWLANProfile

The client can send this request to OM AXI to change a WLAN profile data set. The *id* has to be filled in by the client to identify the record to be changed. Additionally the attributes which have to be changed must be filled in by the client.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
profile	WLANProfileType	yes	Data of WLAN profile to change.

If the change was successful, the reply contains all attributes which *GetWLANProfile* would return. Some data fields may be changed or set by the OMM. If the change fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetWLANProfileResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.45.7 EventWLANProfileCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *WLANProfileCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with only an *id* in the *profile* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
profile	WLANProfileType	yes	Changed attributes
deleted	boolean, default false		„1” or “true”, if this data set has been deleted

Available filters: (none)

4.46 WLAN Access Control List Configuration And Maintenance

If this feature is enabled for a certain WLAN profile (attribute *useACL* of *RFProfileType*), only devices with MAC addresses contained in the access control list are allowed to connect to that profile.

4.46.1 ACLEntryType

This type contains all configuration data fields for an ACL entry.

Name	Type	Description
------	------	-------------

ethAddr	string	Ethernet address, format “00:11:22:aa:bb:cc”. This element is used to identify an entry in the ACL list.
name	string	Optional name of this ACL entry

Note that *ethAddr* is the unique key for these data sets for one WLAN profile. This means it is not only part of the data, but also used to address a certain entry.

4.46.2 CreateACLEntry

With this request the client can create a new entry in the access control list for a given WLAN profile.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
wlanProfile	integer	yes	WLAN profile this request is valid for
entry	ACLEntryType	yes	Entry

Currently it is only allowed to add one entry at once.

If the creation was successful, the reply contains all attributes which *GetACLEntry* would return. Some data fields may be changed or set by the OMM. If the creation fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreateACLEntryResp*. It contains all actual created attributes in addition to the common attributes:

Name	Type	Description
wlanProfile	integer	WLAN profile (when getting multiple entries this is filled in even if no records are returned)
entry	sequence of ACLEntryType	ACL entry records, if found

Possible values for *errCode*:

Value	Description
ENoEnt	The WLAN profile does not exist
ENoMem	List is full

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.46.3 GetACLEntry

With this request the client can acquire one or more ACL entries from the OMM. The request contains a MAC address (*ethAddr*) from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

The responses are ordered by *ethAddr*, starting with the smallest, i.e. “00:30:42:32:10:20” comes before “00:30:50:00:03:03”.

If the client asked for a certain number records and the given *ethAddr* does not exist for the chosen WLAN profile, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given *ethAddr* has to match to a record in this WLAN profile exactly. If there is a record with this *ethAddr*, this record is returned, otherwise *ENoEnt*.

To acquire all data sets for profile 2, one could send *GetACLEntry* (*wlanProfile*="2", *ethAddr* = "00:00:00:00:00:00", *maxRecords* = "20"). If the response contains 20 records, take the last *ethAddr* plus one and send another *GetACLEntry*. Repeat this until you get less than 20 records back.

Note how to increment the addresses: Start with the right hand side number, if there is an overflow, continue with the next one, i.e. "00:30:42:24:FF:FF" plus 1 is "00:30:42:25:00:00".

Following fields are defined for the request *GetACLEntry* in addition to the common attributes:

Name	Type	Mandatory	Description
ethAddr	string	yes	First Ethernet address of ACL entry/entries to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.
wlanProfile	integer	yes	WLAN profile this request is valid for

The reply to this request is an object called *GetWLANProfileResp*. It contains following elements in addition to the common attributes:

Name	Type	Description
wlanProfile	integer	WLAN profile (when getting multiple entries this is filled in even if no records are returned)
entry	sequence of ACLEntryType	ACL entry records, if found

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.46.4 SetACLEntry

With this request the client can change an entry in the access control list for a given WLAN profile.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
wlanProfile	integer	yes	WLAN profile this request is valid for
entry	ACLEntryType	yes	Entry

Currently it is only allowed to add one entry at once.

If the change was successful, the reply contains all attributes which *GetACLEntry* would return. Some data fields may be changed or set by the OMM. If the change fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetACLEntryResp*. It contains all actual set attributes in addition to the common attributes:

Name	Type	Description
wlanProfile	integer	WLAN profile (when getting multiple entries this is filled in even if no records are returned)
entry	sequence of ACLEntryType	ACL entry records, if found

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.46.5 DeleteACLEntry

With this request the client can delete an ACL entry in the OMM. The key is the *wlanProfile* and the *ethAddr*.

Following fields are defined for the request *DeleteACLEntry* in addition to the common attributes:

Name	Type	Mandatory	Description
wlanProfile	integer	yes	WLAN profile this request is valid for
deleteAll	boolean	no	„1” or “true”, to delete all entries in this profile
entry	ACLEntryType	no	Entry to be deleted

If *deleteAll* is set, no entry needs to be given; otherwise exactly one entry has to be in the request.

The reply to this request is an object called *DeleteACLEntry*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

Possible values for *errCode*:

Value	Description
ENoEnt	No record found for given <i>wlanProfile</i> and <i>ethAddr</i>

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.46.6 EventACLCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *ACLCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all entries which have been changed.

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with an *entry* element and the attribute *deleted="1"*. There is a special case: When *deleteAll* is set, all data sets has been removed. In this case the message does not contain these entries.

Attributes sent in this notification:

Name	Type	Mandatory	Description
wlanProfile	integer	yes	WLAN profile this event is valid for
entry	Sequence of ACLEntryType	no	Deleted or created records
deleted	boolean, default false	no	„1” or “true”, if this data set has been deleted
deleteAll	boolean, default false	no	„1” or “true”, if all data set of this WLAN profile have been deleted

Available filters: (none)

4.47 WLAN Connected Clients Monitoring

With these messages the addresses of connected clients can be monitored.

4.47.1 WLANClientType

This type contains all data fields for a connected client.

Name	Type	Description
ethAddr	string	Ethernet address, format “00:11:22:aa:bb:cc”

4.47.2 GetWLANClients

With this request the client can acquire entries from the list of connected clients. The request contains a MAC address (*ethAddr*) from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets is returned, the client knows that it reached end of file.

The responses are ordered by *ethAddr*, starting with the smallest, i.e. “00:30:42:32:10:20” comes before “00:30:50:00:03:03”.

If the given *ethAddr* does not exist for the chosen WLAN profile, subsequent entries will be returned.

To acquire all data sets for *rfpld* 2, one could send *GetWLANClients* (*rfpld="2"*, *ethAddr = "00:00:00:00:00:00"*, *maxRecords = "20"*). If the response contains 20 records, take the last *ethAddr* plus one and send another *GetWLANClients*. Repeat this until you get less than 20 records back.

Note how to increment the addresses: Start with the right hand side number, if there is an overflow, continue with the next one, i.e. “00:30:42:24:FF:FF” plus 1 is “00:30:42:25:00:00”.

Following fields are defined for the request *GetWLANClients* in addition to the common attributes:

Name	Type	Mandatory	Description
ethAddr	string	yes	First Ethernet address of ACL entry/entries to get
maxRecords	integer	yes	Maximal number of records to return. Not more than 20 allowed. The special case 0 is not allowed here.

rfpld	integer	yes	RFP ID this request is valid for
-------	---------	-----	----------------------------------

The reply to this request is an object called *GetWLANProfileResp*. It contains following elements in addition to the common attributes:

Name	Type	Description
rfpld	integer	RFP ID (this is filled in even if no records are returned)
entry	sequence of WLANClientType	Connected WLAN clients, if found

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.47.3 EventWLANClient

This event is sent by OM AXI when a WLAN client connects or disconnects. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *WLANClient*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains an entry of a client which connected or closed a connection.

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with only an *entry* element and the attribute *deleted="1"*. There is a special case: When *deleteAll* is set, all data sets has been removed. In this case the message does not contain these entries.

Attributes sent in this notification:

Name	Type	Mandatory	Description
rfpld	integer	yes	RFP ID this event is valid for
entry	WLANClientType	no	Client which connected or disconnected
deleted	boolean, default false	no	„1” or “true”, if this client is disconnected
deleteAll	boolean, default false	no	„1” or “true”, if all clients of this RFP have been disconnected

Available filters: (none)

4.48 PBX Enrolment Management

On systems which do not support PBX enrolment configuration (see *OpenResp*), the requests *GetPBXEnrolment* etc. are not used. The PBX enrolment is used to retrieve DECT phone enrolment data from a PBX.

Note: The elements “protocol” and “path” of the attribute *URLType* in following requests/responses are not relevant for A5000 OMM systems, they will be ignored and set to ‘protocol=“FTPS”’ and ‘path=“”’ per default, but they must be set in any *SetPBXEnrolment* request somehow!

4.48.1 GetPBXEnrolment

With this request the client can query the *PBXEnrolment* settings.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetPBXEnrolmentResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	no	In case of success: Actually settings to the access the PBX for the DECT phone enrolment file.
hourInterval	integer	no	Time interval in hours to retrieve an update of the DECT phone enrolment file from the PBX.
triggerDateTime	DateTimeType	no	Start point of the hour interval.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.48.2 SetPBXEnrolment

With this request the client can set the PBX enrolment settings.

Following fields are defined for the request *SetPBXEnrolment* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	no	Actually settings to the access the PBX for the DECT phone enrolment file.
hourInterval	integer	no	Time interval in hours to retrieve an update of the DECT phone enrolment file from the PBX.
triggerDateTime	DateTimeType	no	Start point of the hour interval. This attribute must be set when an <i>hourInterval</i> is set. The optional second field will be ignored.

All attributes which have to be changed must be filled in by the client.

The reply is an object called *SetPBXEnrolmentResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.48.3 EventPBXEnrolmentCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	no	Actually settings to the access the PBX for the DECT phone enrolment file, if changed.
hourInterval	integer	no	Time interval in hours to retrieve an update of the DECT phone enrolment file from the PBX, if changed.
triggerDateTime	DateTimeType	no	Start pointer of the hour interval, if changed.

Available filters: (none)

4.49 Site Configuration And Maintenance

4.49.1 SiteType

This type contains all data fields of a Site. It is used in different requests and responses defined in this document.

Name	Type	Description
id	integer	Site ID, numbering starts at 1, 0 is invalid
name	string	Name of the site
nRFPs	integer	Number of RFPs in this site (read only)
wideband	boolean	„1” or “true”, if codec G.722 (Hi-Q audio technology / wide band) for this site is enabled.
dectSecurity	boolean	„1” or “true”, if DECT security for this site is enabled.
videoStreaming	boolean	„1” or “true”, if device video streaming for this site is enabled. Attribute depends on <i>haveVideo</i> attribute in <i>OpenResp</i> .
srtp	string	One of “Only”, “Preferred” or “Disabled” (default). Attribute depends on <i>haveSRTP</i> attribute in <i>OpenResp</i> .

4.49.2 GetSiteSummary

With this request the client can query common information about the sites configured.

Following fields are defined for the request *GetSiteSummary* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to this request is an object called *GetSiteSummaryResp*. It contains following fields in addition to the common attributes:

Name	Type	Description
nRecords	integer	The total number of sites set up.
idFirst	integer	The ID of the first site set up.

rfpSiteNum	integer	Number of sites with RFPs.
g722SiteNum	integer	Number of sites having G.722 enabled.
dectSecuritySiteNum	integer	Number of sites DECT security enabled.
videoStreamingSiteNum	integer	Number of sites video streaming enabled. Attribute depends on <i>haveVideo</i> attribute in <i>OpenResp</i> .
srtplibSiteNum	integer	Number of sites having SRTP enabled. Attribute depends on <i>haveSRTP</i> attribute in <i>OpenResp</i> .

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.49.3 EventSiteSummary

This event is sent by OM AXI when the site summary has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf* or to *SiteSummary*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Description
nRecords	integer	The total number of sites set up.
idFirst	integer	The ID of the first site set up.
rfpSiteNum	integer	Number of sites with RFPs.
g722SiteNum	integer	Number of sites having G.722 enabled.
dectSecuritySiteNum	integer	Number of sites DECT security enabled.
videoStreamingSiteNum	integer	Number of sites video streaming enabled. Attribute depends on <i>haveVideo</i> attribute in <i>OpenResp</i> .
srtplibSiteNum	integer	Number of sites having SRTP enabled. Attribute depends on <i>haveSRTP</i> attribute in <i>OpenResp</i> .

Available filters: (none)

4.49.4 GetSite

With this request the client can acquire one or more Site data sets from the OMM. The request contains an *id* from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

These IDs usually start at 1, but possibly not all numbers are assigned, i.e. the numbering may not be contiguous.

The responses are ordered by *id*, starting with the smallest.

If the client asked for a certain number of records and the given *id* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given *id* has to match to a record exactly. If there is a record with this *id*, this one is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetSite* (*id* = "1", *maxRecords* = "20"). If the response contains 20 records, take the last ID plus one and send another *GetSite*. Repeat this until you get less than 20 records back.

Following fields are defined for the request *GetSite* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	First ID of sites to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to this request is an object called *GetSiteResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
site	sequence of SiteType	no	site records, if available

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.49.5 CreateSite

The client can send this request to OM AXI to create a new site data set.

If there is no *id* assigned by the client or if it is 0, the OMM will choose one. If the client assigns an *id*, it must not exist yet.

Following element is defined for the request *CreateSite* in addition to the common attributes:

Name	Type	Mandatory	Description
site	SiteType	yes	Data of site to create.

A site must always have a *name*.

If the creation was successful, the reply contains all attributes which *GetSite* would return. Some data fields may be changed or set by the OMM. If the creation fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreateSiteResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.49.6 DeleteSite

With this request the client can delete a site record in the OMM. The key is the *id*.

Following fields are defined for the request *DeleteSite* in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

id	integer	yes	ID of site to be deleted
----	---------	-----	--------------------------

Note that the last site cannot be deleted, i.e. there must always be at least one site.

Sites which still have RFPs attached cannot be deleted.

The reply to this request is an object called *DeleteSiteResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

Possible values for *errCode*:

Value	Description
EFailed	This is the last site, it cannot be deleted
EExist	There are still RFPs in this site, it cannot be deleted

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.49.7 SetSite

The client can send this request to OM AXI to change a site data set. The *id* has to be filled in by the client to identify the record to be changed. Additionally the attributes which have to be changed must be filled in by the client.

Following element is defined for the request *SetSite* in addition to the common attributes:

Name	Type	Mandatory	Description
site	SiteType	yes	Data of site to change.

If the change was successful, the reply contains all attributes which *GetSite* would return. Some data fields may be changed or set by the OMM. If the change fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetSiteResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.49.8 EventSiteCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SiteCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with only an *id* in the *site* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
site	SiteType	yes	Changed site, only changed attributes of the site are sent
deleted	boolean, default false	no	„1” or “true”, if this data set has been deleted

Available filters: (none)

4.50 Digit Treatment Configuration And Maintenance

4.50.1 SiteIdType

This type contains the data fields of a site used as sequence in the *DigitTreatmentType*. It is used in *digitTreatment* requests and responses defined in this chapter.

Name	Type	Description
siteld	integer	ID of a site

4.50.2 DigitTreatmentType

This type contains all data fields of a digit treatment rule. It is used in different requests and responses defined in this document.

Depending on the *OpenResp* the *DigitTreatmentType* is structured either for the parameter “haveDigitTreatment=1” as follows:

Name	Type	Description
id	integer	Unique ID for this data set start at 0, i.e. -1 is an invalid ID.
externalPattern	string	External numbers to be replaced or scanned.
internalPattern	string	Internal numbers to be replaced or scanned
directory	boolean, default false	If true, rules applies to corporate directory entries
direction	enumeration	One of “Incoming”, “Outgoing”, “Both” “Incoming”: Rule applies to the calling party number of an incoming call “Outgoing”: Rule applies to the dialled number of an outgoing call “Both”: Rule applies to the calling party number of an incoming call and the reverse rule to the dialled number of an outgoing call “Directory”: Rule applies to corporate directory (LDAP) entries
sites	sequence of SiteIdType	One or more <i>SiteIdTypes</i> , if missing the rule applies to all sites i.e. the rule will be applied to all calls or corporate directory entries.

or for the parameter “haveDigitTreatmentSimple=1” as follows:

Name	Type	Description
------	------	-------------

id	integer	Unique ID for this data set start at 0, i.e. -1 is an invalid ID.
prefix	string	Prefix to be replaced
substitute	string	Substitute

Which fields are mandatory depend on OMM type, request and response type and permissions of the user being logged in.

4.50.3 CreateDigitTreatment

The client can send this request to OM AXI to create a new digit treatment rule.

If there is no *id* assigned by the client, the OMM will choose one. If the client assigns an *id*, it must not exist yet.

Currently the given *id* is always ignored.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
rule	DigitTreatmentType	yes	Data of digit treatment rule to create

If the creation was successful, the reply contains all attributes which *GetDigitTreatment* would return. Some data fields may be changed or set by the OMM. If the creation fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreateDigitTreatmentResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.50.4 DeleteDigitTreatment

With this request the client can delete a digit treatment rule in the OMM. The key is its *id*.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	ID of record to be deleted

The reply to this request is an object called *DeleteDigitTreatmentResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.50.5 GetDigitTreatment

With this request the client can acquire one or more digit treatment rules from the OMM. The request contains an ID from which starting subsequent digit treatment data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

The *ids* usually start at 0, but possibly not all *ids* are assigned, i.e. the numbering may not be contiguous.

The responses are ordered by *id*, starting with the smallest.

If the client asked for a certain number records but the given *id* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given *id* has to match to a record exactly. If there is a record with this *id*, it is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetDigitTreatment* (*id* = "0", *maxRecords* = "20"). If the response contains 20 records, take the last *id* plus one and send another *GetDigitTreatment*. Repeat this until you get less than 20 records back.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	First ID of records to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to *GetDigitTreatment* is an object called *GetDigitTreatmentResp*. It contains following element in addition to the common attributes:

Name	Type	Description
rule	sequence of DigitTreatmentType	Digit treatment records, if found

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.50.6 GetDigitTreatmentSummary

This request is sent from the client to the OMM.

With this request the client can query common information about the digit treatment rules configured.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to this request is an object called *GetDigitTreatmentSummaryResp*. It contains following fields in addition to the common attributes:

Name	Type	Description
nRecords	integer	The total number of records set up
idFirst	integer	The ID of the first record set up

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.50.7 SetDigitTreatment

The client can send this request to OM AXI to change a digit treatment rule. The *id* has to be filled in by the client to identify the record to be changed. Additionally the attributes which have to be changed must be filled in by the client.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
rule	DigitTreatmentType	yes	Data of record to be changed

If the change was successful, the reply contains all attributes which *GetDigitTreatment* would return. Some data fields may be changed or set by the OMM. If the change fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetDigitTreatmentResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

Value	Description
EFailed	This device/attribute cannot be changed (e. g. because of relation type)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.50.8 EventDigitTreatmentCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *DigitTreatmentCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains the rule which have been changed.

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with only an *id* in the *rule* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
rule	DigitTreatmentType	yes	Changed rule
deleted	boolean, default false	no	„1” or “true”, if this data set has been deleted

Available filters: (none)

4.51 Feature Access Codes Configuration And Maintenance

Feature Access Codes (FAC) are numbers which can be dialed from a phone to invoke certain features.

4.51.1 FACType

This type contains all attributes of a FAC.

It has following attributes:

Name	Type	Description
feature	FeatureType	The feature invoked by this FAC
enable	boolean	„1” or “true”, if this FAC is enabled
fac	string	The code, a dial able number

FeatureType can have following values:

Value	Description
ActivateSubscription	Activate the subscription mode
ActivateWildcard	Activate wildcard subscription mode
DeactivateSubscription	Deactivate subscription mode
UserLogin	User Login
UserLogout	User Logout
SystemCredentialPasswd	Change the system credential password
BlindTransfer	Blind transfer

UserLogin, *UserLogout*, *PINChange* are not available on all OMM versions. *GetFACList* can be used to find out which FACs are supported.

4.51.2 GetFACList

A client can get a list with all FACs configured using this request.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(none)

The reply to this request is an object called *GetFACListResp*. It contains following element in addition to the common attributes:

Name	Type	Description
fac	sequence of FACType	All FAC records

Possible values for *errCode*:

Value	Description
-------	-------------

ENoEnt	No FAC exists
--------	---------------

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.51.3 SetFAC

A client can change a FAC using this request.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
fac	FACType	yes	FAC to be changed

Only one FAC can be changed at once. However, the response contains the full, updated list of FACs in case of success.

The reply to this request is an object called *SetFACResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

Value	Description
EExists	The FAC already exists

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.51.4 SetFACList

A client can change one or up to all FACs using this request.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
fac	sequence of FACType	yes	List of all FAC to be changed or to stay unchanged.

The response contains the full, updated list of all FACs in case of success.

The reply to this request is an object called *SetFACListResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.51.5 GetFACPrefix

A client can get global FAC prefix with this request.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(none)

The reply to this request is an object called *GetFACResp*. It contains following element in addition to the common attributes:

Name	Type	Description
prefix	string	FAC prefix

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.51.6 SetFACPrefix

A client can change the global FAC prefix using this request.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
prefix	string	yes	New FAC prefix

The reply to this request is an object called *SetFACResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.51.7 EventFACCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *FACCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with only an *id* in the *fac* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
fac	FACType	yes	Changed attributes
deleted	boolean, default false		„1” or “true”, if this data set has been deleted

Available filters: (none)

4.51.8 EventFACPrefixCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *FACConf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
prefix	string	yes	Changed prefix

Available filters: (none)

4.52 Corporate Directory Configuration And Maintenance

4.52.1 LDAP Corporate Directories

4.52.1.1 LDAPType

This type contains all attributes needed to configure an LDAP server.

Depending on the *OpenResp* the *LDAPType* is structured either for the parameter "haveLDAP=1" in the *OpenResp* as follows:

Name	Type	Mandatory Create/Set	Description
id	integer	no/yes	Identification of the LDAP data set. This value is delivered by the server in a <i>CreateLDAPResp</i>
name	string	yes/no	LDAP data set name description
active	boolean	yes/no	„1" or "true", if this data set is active
order	integer	no/no	Order number to prioritize this LDAP directory data set. This setting might affect the order number in other corporate directory data sets. A reordering of all corporate directory data sets will be performed. If empty, the LDAP directory data set is queued as last entry of the all corporate directory data sets. See also at the <i>corpDirOrder</i> number of the XML application directories in 4.60.1.
server	string	yes/no	LDAP server name or IP address
port	integer	yes/no	LDAP server port
searchBase	string	yes/no	Search base To make search requests unique for different users the search base configuration can include placeholders which are replaced by user specific values when submitting the LDAP request to a server. The following placeholders are defined: "<TEL>" which is

			replaced by the specific telephone number of the user, "<DESC1>" which is replaced by the "hierarchy1" attribute value of the user "<DESC2>" which is replaced by the "hierarchy2" attribute value of the user (see 4.39.1). Note: The telephone number in SIP - DECT is not limited to numeric character.
username	string	no/no	User name for server access
password	string	no/no	Password for server access, encrypted with public key
searchType	enumeration	yes/no	One of "GN" (given name), "SN" (surname), "CN" (common/complete name). Note: Up to now CN is not supported
displayType	enumeration	yes/no	One of "CN", "SN, GN"
timeout	integer	yes/no	Search timeout in seconds

The search type "Name" means search for *surname*, "Full" means search for *given name*.

The display type "Name" means display *Surname, given name*, "Full" means *Given name and surname*.

4.52.1.2 CreateLDAP

With this request the client can create a corporate directory data set. The OMM supports a limited number of LDAP data sets.

Following fields are defined for the request *CreateLDAP* in addition to the common attributes:

Name	Type	Mandatory	Description
ldap	LDAPType	yes	LDAP configuration

If the creation was successful, the reply contains all attributes which *GetLDAP* would return.

The reply is an object called *CreateLDAPResp*. It contains the created LDAP data set with actual set attributes.

Possible values for *errCode*:

Value	Description
EExists	The name of the LDAP data set already exists

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.52.1.3 DeleteLDAP

With this request the client can delete an LDAP data set.

Following fields are defined for the request *DeleteLDAP* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	Identification of the LDAP data set. This value is delivered by the server in a <i>CreateLDAPResp</i>

The reply to this request is an object called *DeleteLDAPResp*. It contains following fields in addition to the common attributes:

			(no additional fields defined)
--	--	--	--------------------------------

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.52.1.4 GetLDAP

With this request the client can query the configuration of the corporate directory.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	no	<i>Only supported, when the parameter "haveLDAP=1" is set in the OpenResp.</i> Identification of the LDAP data set. This value is delivered by the server in a <i>CreateLDAPResp</i> . If empty, all data sets will be delivered.

The reply is an object called *GetLDAPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
ldap	sequence of LDAPType	no	LDAP configuration

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.52.1.5 SetLDAP

With this request the client can configure change a corporate directory data set.

Only attributes which have to be changed need to be specified.

Following fields are defined for the request *SetLDAP* in addition to the common attributes:

Name	Type	Mandatory	Description
ldap	LDAPType	yes	LDAP configuration

If the change was successful, the reply contains all attributes which *GetLDAP* would return.

The reply is an object called *SetLDAPResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.52.1.6 EventLDAPCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
ldap	LDAPType	yes	LDAP configuration
deleted	boolean, default false	no	„1” or “true”, if this data set has been deleted

Available filters: (none)

4.52.2 XML Corporate Directories

The XML corporate directories are configured within the XML application configuration and maintenance chapter. Please refer to 4.60 how XML corporate directories can be created, deleted got, set and notified.

4.53 Alerting Configuration And Maintenance

4.53.1 AlarmTriggerType

This type contains all data fields of an alarm trigger. It is used in different requests and responses defined in this document.

Name	Type	Description
id	integer	Alarm trigger ID, starts at 0, -1 is invalid
triggerId	string	Textual identifier of the alarm trigger, not case sensitive
fac	string	Feature access code, not case sensitive
comment	string	Comment
num	string	Phone number to be called

The attribute, *fac* must be unique.

There are some special alarm triggers with the *fac* and *triggerId* set to the magic words “SOS” and “MANDOWN”, “. These triggers refer to special events, they cannot be deleted and their *fac* and *triggerId* cannot be modified.

There are some pre-defined alarm triggers which use reserved words:

<i>fac, triggerId</i>	Description
SOS	The SOS event is triggered by a DECT phone with a configured SOS number, when a “SOS” is identified at this DECT phone. This trigger cannot be deleted. Only the <i>comment</i> and <i>num</i> may be changed.
MANDOWN	The MANDOWN event is triggered by a DECT phone with a configured MANDOWN number, when a “MANDOWN” is identified at this DECT phone. This trigger cannot be deleted. Only the <i>comment</i> and <i>num</i> may be changed.
PAGEBYMENU	The PAGEBYMENY event is triggered by a DECT phone, when this DECT phone initiates a paging request in the system menu. This trigger cannot be deleted.

OMM-...	Health state alarms, see <i>EventAlarmTrigger</i> (4.53.8)
UMON-WARN-USERSTATE	The UMON-WARN-USERSTATE event is triggered when this DECT phone escalate the detection of the user unavailability.
UMON-ERR-USERSTATE	The UMON-ERR-USERSTATE event is triggered when this DECT phone escalate the final detection of the user unavailability.
UMON-OK-USERSTATE	The UMON-OK-USERSTATE event is triggered when this DECT phone detects the user availability.
LOC-ERR-USERSTATE	The LOC-ERR-USERSTATE event is triggered when this DECT phone escalate the final detection of the user unavailability and related configuration is done.

4.53.2 GetAlarmTriggerSummary

With this request the client can query common information about the alarm triggers configured.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to this request is an object called *GetAlarmTriggerSummaryResp*. It contains following fields in addition to the common attributes:

Name	Type	Description
nRecords	integer	The total number of alarm triggers set up.
idFirst	integer	The ID of the first alarm trigger set up.

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.53.3 GetAlarmTrigger

With this request the client can acquire one or more alarm trigger data sets from the OMM. The request contains an *id* from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

These IDs usually start at 0, but possibly not all numbers are assigned, i.e. the numbering may not be contiguous.

The responses are ordered by *id*, starting with the smallest.

If the client asked for a certain number of records and the given *id* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given *id* has to match to a record exactly. If there is a record with this *id*, this one is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetAlarmTrigger* (*id* = "0", *maxRecords* = "20"). If the response contains 20 records, take the last ID plus one and send another *GetAlarmTrigger*. Repeat this until you get less than 20 records back.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	First ID of alarm triggers to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to this request is an object called *GetAlarmTriggerResp*. It contains following element in addition to the common attributes:

Name	Type	Description
trigger	sequence of AlarmTriggerType	alarm trigger records, if found

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.53.4 CreateAlarmTrigger

The client can send this request to OM AXI to create a new alarm trigger data set.

If there is no *id* assigned by the client or if it is -1, the OMM will choose one. If the client assigns an *id*, it must not exist yet.

Following element is defined for the request *CreateAlarmTrigger* in addition to the common attributes:

Name	Type	Mandatory	Description
trigger	AlarmTriggerType	yes	Data of site to create.

If the creation was successful, the reply contains all attributes which *GetAlarmTrigger* would return. Some data fields may be changed or set by the OMM. If the creation fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

Note that *triggerIds* and *facts* “SOS” and “MANDOWN” are reserved and cannot be configured.

The reply to this request is called *CreateAlarmTriggerResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.53.5 DeleteAlarmTrigger

With this request the client can delete an alarm trigger record in the OMM. The key is the *id*.

Following fields are defined for the request *DeleteAlarmTrigger* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	ID of alarm trigger to be deleted

The reply to this request is an object called *DeleteAlarmTriggerResp*. It contains following fields in addition to the common attributes:

			(no additional fields defined)
--	--	--	--------------------------------

Possible values for *errCode*:

Value	Description
EFailed	This alarm trigger must not be deleted (“SOS”, “MANDOWN”)

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.53.6 SetAlarmTrigger

The client can send this request to OM AXI to change an alarm trigger data set. The *id* has to be filled in by the client to identify the record to be changed. Additionally the attributes which have to be changed must be filled in by the client.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
trigger	AlarmTriggerType	yes	Data of alarm trigger to change.

If the change was successful, the reply contains all attributes which *GetAlarmTrigger* would return. Some data fields may be changed or set by the OMM. If the change fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

Note that *triggerIds* and *facs* for “SOS” and “MANDOWN” are reserved.

The reply to this request is called *SetAlarmTriggerResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

Value	Description
EFailed	Failed to change a protected attribute (“SOS”, “MANDOWN”)

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.53.7 EventAlarmTriggerCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *AlarmTriggerCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with only an *id* in the *alarmTrigger* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
alarmTrigger	AlarmTriggerType	yes	Changed attributes

deleted	boolean, default false	no	„1” or “true”, if this data set has been deleted
---------	------------------------	----	--

Available filters: (none)

4.53.8 EventAlarmTrigger

This event is sent by OM AXI if there is an alarm of the subscribed trigger type. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *AlarmTrigger*.

DECT phones can be the source of user caused alarms like SOS, Man Down or FAC/menu triggered alarms.

RFPs and OMMs can be the source of automatically generated alarms concerning certain malfunctions. These health state alarms have some pre-defined values for *trigger*. They start with “OMM-”, followed by one of the severities:

“OK-”, “IDLE-”, “WARNING-”, “ERROR-”

One of the causes:

“SYNC”, “STANDBY”, “DBTRANSFER”, “DOWNLOAD”, “RFP”, “LICENSE”, “IMA”, “UMO”, “UDS”, “USBMEM”, “LICENSESERVER”, “BLUETOOTH”, “VIDEO”, “SIP”, “SIPCERT”, “AXIPROVISIONINGCOMMANDS”, “PROVISIONINGSERVER”, “DHCPSEVER”, “USERDEVICESYNC”

And optionally followed by a reason string:

“-<specific reason>” // formatted in upper case letters and blanks represented by a ‘-’ character

For example *trigger*="OMM-WARNING-SYNC", for details about health states refer to Remote System Dump Configuration And Maintenance

ActivateRemoteSystemDump

This request is sent from the client to the OMM.

With this request the client can activate a remote system dump to the configured destination.

Following fields are defined for the request *ActivateRemoteSystemDump* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetRemoteSystemDump* is an object called *ActivateemoteSystemDumpResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The client needs one of these permissions to use this request: *AllCnfRead*

4.53.9 GetRemoteSystemDump

This request is sent from the client to the OMM.

With this request the client can query for the settings of the remote system dump to the configured destination.

Following fields are defined for the request *GetRemoteSystemDump* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetRemoteSystemDump* is an object called *GetRemoteSystemDumpResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, system dump shall be uploaded to a remote server daily on time.
hour	integer	yes	Hour time value to upload the remote system dump.
minute	integer	yes	Minute time value to upload the remote system dump.
url	URLType	yes	Server configuration settings where the remote system dump to be put.

The URL contains the path (directory) where the database file (“sys_dump.txt.gz”) be put to.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.53.10 SetRemoteSystemDump

This request is sent from the client to the OMM.

With this request the client can set the settings of the remote system dump to the configured destination.

Following fields are defined for the request *SetRemoteSystemDump* in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, system dump shall be uploaded to a remote server daily on time.
hour	integer	no	Hour time value to upload the remote system dump.
minute	integer	no	Minute time value to upload the remote system dump.
url	URLType	no	Server configuration settings where the remote system dump to be put.

The URL contains the path (directory) where the database file (“sys_dump.txt.gz”) be put to.

The reply to *SetRemoteSystemDump* is an object called *SetRemoteSystemDumpResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.53.11 EventRemoteSystemDumpCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, system dump shall be uploaded to a remote server daily on time.
hour	integer	no	Hour time value to upload the remote system dump.
minute	integer	no	Minute time value to upload the remote system dump.
url	URLType	no	Server configuration settings where the remote system dump to be put.

Available filters: (none)

Health State Monitoring (4.19).

Attributes sent in this notification:

Name	Type	Mandatory	Description
sendTime	unsigned integer	yes	Original send time in seconds since 01.01.1970 00:00, local time of the DECT system (note that the DECT system and the AXI client might be in different time zones – <i>sendTime</i> will not be adjusted to UTC).
id	unsigned integer, default 0	no	Alarm ID
trigger	string	yes	Alarm trigger identifier
ppn	integer	no	If applicable: PPN of the DECT phone which triggered the alarm
uid	integer	no	If applicable: user ID of the user which triggered the alarm
fromAddr	string	yes	Sender address. In URI format, e. g. “tel:4711”
toAddr	string	no	If applicable: Destination for alarm call, e. g. “tel:110”
dialSuffix	string	no	Additional alarm trigger information for the application given as dial suffix.
content	string	no	Optional additional info about this alarm, always text/plain encoded in UTF-8

Available filters: *ppn*, *rfp*, *omm*, *trigger*

Note: The filters *trigger*, *ppn*, *rfp* cannot be used to select health state alarm triggers (“OMM-...”). If you want to get them, use the filter *omm* instead.

Note: Setting a filter to a certain *rfp* does cause automatically generated health state alarm triggers (“OMM-...”) to be delivered for this RFP. It does not implicitly mean that alarms from DECT phones being at this RFP will be notified.

4.53.12 GenerateHealthStateAlarmTriggers

With this request the client can trigger the OMM to send out all currently present health state alarm triggers. After a positive response all current health state alarms are sent as *EvenAlarmTrigger*. Note the conditions for *EvenAlarmTrigger* in 4.53.8.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to this request is an object called *GenerateHealthStateAlarmTriggersResp*. It contains following element in addition to the common attributes:

Name	Type	Description
		(no additional fields defined)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.53.13 EventAlarmCallProgress

This event is sent by OM AXI when the state of an alarm call changes. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *AlarmCallProgress*.

DECT phones can be the source of user caused alarms like SOS, Man Down or FAC/menu triggered alarms.

Attributes sent in this notification:

Name	Type	Mandatory	Description
ppn	integer	yes	DECT phone which triggered the alarm call
trigger	string	yes	Alarm trigger identifier
id	unsigned integer, default 0	no	Alarm ID
destAddr	string	yes	URI of callee who accepted the call, e. g. "tel:3820"
state	integer	yes	State of alarm call, SIP status codes: 180 Ringing, 200 OK, 404 Not found etc.

Available filters: ppn, trigger

4.54 Remote System Dump Configuration And Maintenance

4.54.1 ActivateRemoteSystemDump

This request is sent from the client to the OMM.

With this request the client can activate a remote system dump to the configured destination.

Following fields are defined for the request *ActivateRemoteSystemDump* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetRemoteSystemDump* is an object called *ActivateemoteSystemDumpResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The client needs one of these permissions to use this request: *AllCnfRead*

4.54.2 GetRemoteSystemDump

This request is sent from the client to the OMM.

With this request the client can query for the settings of the remote system dump to the configured destination.

Following fields are defined for the request *GetRemoteSystemDump* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetRemoteSystemDump* is an object called *GetRemoteSystemDumpResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, system dump shall be uploaded to a remote server daily on time.
hour	integer	yes	Hour time value to upload the remote system dump.
minute	integer	yes	Minute time value to upload the remote system dump.
url	URLType	yes	Server configuration settings where the remote system dump to be put.

The URL contains the path (directory) where the database file (“sys_dump.txt.gz”) be put to.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.54.3 SetRemoteSystemDump

This request is sent from the client to the OMM.

With this request the client can set the settings of the remote system dump to the configured destination.

Following fields are defined for the request *SetRemoteSystemDump* in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

enable	boolean	no	„1” or “true”, system dump shall be uploaded to a remote server daily on time.
hour	integer	no	Hour time value to upload the remote system dump.
minute	integer	no	Minute time value to upload the remote system dump.
url	URLType	no	Server configuration settings where the remote system dump to be put.

The URL contains the path (directory) where the database file (“sys_dump.txt.gz”) be put to.

The reply to *SetRemoteSystemDump* is an object called *SetRemoteSystemDumpResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.54.4 EventRemoteSystemDumpCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, system dump shall be uploaded to a remote server daily on time.
hour	integer	no	Hour time value to upload the remote system dump.
minute	integer	no	Minute time value to upload the remote system dump.
url	URLType	no	Server configuration settings where the remote system dump to be put.

Available filters: (none)

4.55 Messaging

4.55.1 MessageType

This type contains all fields of a message. It is used in different requests and notifications. All of these fields are attributes or sequences of elements.

Name	Type	Description
sendTime	unsigned integer	Original send time in seconds since 01.01.1970 00:00 as it shall be used to display the send time at the DECT phone (note that the DECT phone and the AXI client might be in

		different time zones – this will not be adjusted for the DECT phone).
id	unsigned integer	Message ID. <i>sendTime</i> and <i>id</i> together should be chosen to be unique. This may be left out if the message does not need to be identified later. The default setting is 0.
ppn	integer	If sent from a DECT phone: PPN
fromAddr	string	Sender address. In URI format, e. g. “mailto:foo@bar.org”, “tel:4711”. The maximal length is 64 bytes/characters(UTF-8 characters might be more than 1 byte). The client can feel free to use this address. An address like „alarmserver:alarmtype“ would be correct. It is up to the client use this address on how messages handled internally.
fromName	string	Human readable name of the sender, if available. The maximal length is 64 bytes/characters(UTF-8 characters might be more than 1 byte).
toAddr	string	Recipient address in URI format. This may also be an explicit PPN, e. g. “ppn:23” to address a physical DECT phone. The maximal length is 128 bytes/characters(UTF-8 characters might be more than 1 byte).
toName	string	Human readable name of the recipient, if available The maximal length is 64 bytes/characters(UTF-8 characters might be more than 1 byte).
callbackAddr	string	Callback address in URI format. This must be an explicit “cb:<calling party number>” to address a physical DECT phone. The maximal length is 64 bytes/characters(UTF-8 characters might be more than 1 byte).
callbackName	string	Human readable callback name of the callback receiver, if available. The maximal length is 64 bytes/characters(UTF-8 characters might be more than 1 byte).
autoCallback	boolean	Automatic call establishment from the recipient DECT phone of the message to the callback address party. Default=False Note: <i>autoCallback</i> is not supported to a video device.
priority	enumeration, default <i>Normal</i>	One of “Info”, “Low”, “Normal”, “High”, “Emergency”, “LocatingAlert”.
folder	FolderType, default “Inbox”	Where to store the message.
noReply	boolean, default false	„1” or “true”, if the receiver must not reply to this message
autoDelete	boolean, default false	„1” or “true”, if the message has to be deleted after it reached its final state (i.e. all required confirmations sent)
popUp	boolean, default false	„1” or “true”, for additional pop-up window

encoding	enum, default "UTF-8"	One of "UTF-8"
contentType	enum, default "text/plain"	One of "text/plain", "text/x-vcard". For vCard see comments below.
confirm	sequence of MessageConfirmType	Confirmations and their choices wanted for this message.
content	string	Message content.
melody	MelodyType	Optional melody signalling parameter when the message is received at the DECT phone.
explicitToneSelection	string	Optional explicit tone selection to be signaled. The string values of the tone to be sound is specified the table further below and depends on the DECT phone model. If this element is used the <i>melody</i> element is obsolete. Depending on the DECT phone model not all the strings may work at the DECT phone. The string value is not check for correctness. Therefore wrong or unknown string values are ignored. Example values see table below.
signallingVolume	integer	Optional acoustical signalling volume parameter when the message is received at the DECT phone (value 0/local volume or 1 till 100% volume), default is 0.
increasingVolume	boolean	„1" or "true", if increasing volume shall be active, when the message is received at the DECT phone. Otherwise the local DECT phone setting is used. The default setting is "false". Note: Increasing volume does not work with all used tones.
vibraCall	boolean	„1" or "true", if vibration signalling shall be active, when the message is received at the DECT phone. Otherwise the local DECT phone setting is used. The default setting is "false".
discCallOnRecv	boolean	Optional parameter to set the DECT phone into idle state prior (e. g. an established call will be disconnected) when the message is received.
noInbandSignalling	boolean	„1" or "true", if no inband signalling shall be active, when the message is received at the DECT phone. Otherwise the local DECT phone setting is used. The default setting is "false".
ringerTone		„1" or "true", if ringer tone shall be active, when the message is received at the DECT phone. Otherwise the local DECT phone setting is used. The default setting is "false".
textColourR	integer	Optional red text message colour parameter when the message is received at the DECT phone (value -1/no red background colour or 0 till 255), default is -1. See colour note below.
textColourG	integer	Optional green text message colour parameter when the message is received at the DECT phone (value -1/no green background colour or 0 till 255), default is -1.

		See colour note below.
textColourB	integer	Optional blue text message colour parameter when the message is received at the DECT phone (value -1/no blue background colour or 0 till 255), default is -1. See colour note below.
bgColourR	integer	Optional red background message colour parameter when the message is received at the DECT phone (value -1/no red text colour or 0 till 255), default is -1. See colour note below.
bgColourG	integer	Optional green background message colour parameter when the message is received at the DECT phone (value -1/no green text colour or 0 till 255), default is -1. See colour note below.
bgColourB	integer	Optional blue background message color parameter when the message is received at the DECT phone (value -1/no blue text colour or 0 till 255), default is -1. See colour note below.

Notes:

- The bold marked elements are mandatory for a *SendMessage* request.
- The grey marked attributes are only used when the *OM Messaging & Alerting System License* is available, otherwise they will be ignored.
- The maximal length of the attributes *fromAddr*, *toAddr*, *fromName* and *toName* is 128 bytes/characters (UTF-8 characters might be more than 1 byte). An error report for the attribute with the most characters will be done.
Depending on the DECT phone capability and version some attributes might not be supported, in this case they are ignored.
- The client is responsible for all of the message **colour settings** related to a suitable presentation at the DECT phone display (e. g. black text on black background!). The OMM would not correct these settings to make the message readable for the user!
- vCard contents have to be coded according to *vCard Version 3* respectively RFC 2425 and 2426. The DECT phone supports the following fields:

DECT phone-entry-key	key 1	key 2	key 3	key 4	Content
Name	FN	N			<name in utf8/latin1>
Private number	TEL;HOME	...;HOME;VOICE	TEL;ISDN		<number in ascii>
Business number	TEL;WORK	...;WORK;VOICE	TEL;VOICE	TEL;PREF	<number in ascii>
Mobile number	TEL;CELL	...;CELL;VOICE			<number in ascii>

Fax number	TEL;FAX	...;FAX			<number in ascii>
E-Mail	EMAIL	EMAIL;PREF;INTEGER RERNET			<email in ascii>
Quick-dial	X-QC				2...9
Melody name	X-MEL				<xls source name>
VIP-number	X-VIP				<number in ascii>
Character set	VERSION	CHARSET			Char mapper
Framing	BEGIN:VCARD	END:VCARD			

FolderType is an enumeration value defined as follows:

Value	Description
None	Do not store the message, must be combined with <i>popUp="1"</i>
Idle	Show the message in Idle display only
Inbox	Inbox
AddToInbox	Add to inbox silently
AddToPreDefined	Add this message to the pool of pre-defined messages
AddToOutbox	Add this message to the Outbox

MelodyType is an enumeration value defined as follows:

Value	Description
None	No melody (default)
MsgMelody1	Message melody 1 for internal DECT phone melody
MsgMelody2	Message melody 2 for internal DECT phone melody
MsgMelody3	Message melody 3 for internal DECT phone melody
MsgMelody4	Message melody 4 for internal DECT phone melody
MsgMelody5	Message melody 5 for internal DECT phone melody
MsgMelody6	Message melody 6 for internal DECT phone melody
MsgMelody7	Message melody 7 for internal DECT phone melody
MsgMelody8	Message melody 8 for internal DECT phone melody
MsgMelody9	Message melody 9 for internal DECT phone melody
MsgMelody10	Message melody 10 for internal DECT phone melody

Example string values for the *explicitToneSelection* element. For more information and availability see DECT phone documentation:

Value
VIP
Extern
Intern
SOS
Alarm
Appointment
Normal message
Urgent message
Alarm message
Special
CCBS
Recall
Anonymous
Weekend
Butterfly
Barock
Ballade
Fancy
Comelody
Easy groove
Happy fair
Kitafun
Latin dance
Little asia
Mango selassi

Parka
Remember
Rocky lane
Ringing 1
Ringing 2
Ringing 3
Ringing 4
Ringing 5
Ringing 6
Ringing 7
Ring vintage
Vibes
Attack
Doorbell
Boogie
Polka
Classical 1
Classical 2
Classical 3
Classical 4
Alla turca
Entertainer
Jollygood
In the saints
Drunken sailor
Mary had
Shell be walking

Pippi longstocking
Policehorn
Synthesizer
After work
Beep
Basic 1
Basic 2
Basic 3
Basic 4
Basic 5
Basic 6
Basic 7
Basic 8
Alarm 1
Alarm 2
Alarm 3
Alarm 4
Alarm 5
Alarm 6
Alarm 7
6700 One
6700 Two
6700 Three
6700 Four
6700 Five
1 Attention tone
2 Attention tones

3 Attention tones
4 Attention tones
5 Attention tones
6 Attention tones
7 Attention tones
8 Attention tones
9 Attention tones

4.55.2 MessageConfirmType

This type contains information about a message confirmation. If this is contained in a message confirmation request, there may be several choices possible, e. g. “Yes” **and** “No”. If it is used in a confirmation, only the choice done by a user is contained.

MessageConfirmType is an enumeration value defined as follows:

Value	Description
ReadYes	Message was opened for reading
ReadNo	Message was deleted without reading it.
OrderOk	Order accepted
OrderNok	Order not accepted
OrderDontKnow	Not sure whether the order could be accomplished
CompleteDone	Accomplished
CompleteNotDone	Not accomplished
CompleteFailed	Failed to accomplish order

Up to three choices per state (meaning Read, Order or Completion) may be combined.

4.55.3 SendMessage

A client can send this request to send a message to a DECT phone. Consider the license restrictions for this command (see 5.5).

If the message is accepted by the OMM, it is put into a send queue. The response to this request is always sent immediately, but it will take a while until the message can be forwarded to a DECT phone. For that reason the progress of sending the message is reported using Events, refer to 4.55.7.

There are following attributes and elements in addition to the common attributes in this request:

Name	Type	Mandatory	Description
------	------	-----------	-------------

alwaysIndirect	boolean	no	"1" or "true" when this message must always be routed to a messaging application. Note: This parameter is reserved for Mitel SIP-DECT internal use and certain behavior. So, it has not to be used by other clients.
msg	MessageType	yes	Message to be sent.

Normally, when a message is addressed to "ppn:" or "tel:" the OMM forwards the message directly to a DECT phone. However, if the attribute *alwaysIndirect* is set, the message is sent to a messaging application first. In this case the messaging application will take care for the queuing and event notifications. In this case the events for message progress and confirmation explained in the subsequent chapters do not apply.

The locating application uses the scheme "loc:" in the sender URI. This can be used to reply to the locating application and must not be used in other applications.

The OMM will reply positively (i.e. no error) to this request if the message could be queued resp. sent to a messaging application.

The reply to *SendMessage* is an object called *SendMessageResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

Possible values for *errCode*:

Value	Description
ENoEnt	No DECT phone record found for given destination.
ETooLong	Either a string is too long or the length sum of the attributes <i>fromAddr</i> , <i>toAddr</i> , <i>fromName</i> and <i>toName</i> is too long.
ENoMem	The queue for this DECT phone is full. The client may try again after getting an <i>EventMessageQueueEmpty</i> for this DECT phone.
EFailed	If the recipient is a DECT phone, it is not able to receive messages (e. g. hardware limitations). Otherwise this means that currently there is no client connected which is able to receive this address scheme.
ELicense	If the recipient is a DECT phone and the message receive license is not set.

For further possible *errCode* values see 3.3.

For priority *Info* the client needs the permission to use this request: *InfoMessaging*

For priority *Low*, *Normal* and *High* the client needs the permissions to use this request: *Messaging*, *InfoMessaging*

For priority *Emergency* and *LocatingAlert* the client needs the permissions to use this request: *Messaging*, *InfoMessaging*, *Locating*

4.55.4 DeleteMessage

A client can send this request to delete a message from a DECT phone. If the message which has to be deleted is still buffered in the OMM, it is removed from the queue, in this case this request has the same semantic as *CancelMessage*.

If the request is accepted by the OMM, it is put into the same queue as a message which had to be sent would have been put. The response to this request is always sent immediately, but it will take a while until the delete request can be actually forwarded to a DECT phone. For that reason the progress of deleting a message is reported using Events, refer to 4.55.7.

Following fields are defined for the request *DeleteMessage* in addition to the common attributes:

Name	Type	Mandatory	Description
sendTime	unsigned integer	yes	Original send time of the message to be deleted.
id	unsigned integer	yes	Message ID.
toAddr	string	yes	Recipient address of the message to be deleted. Must have the same scheme as the original message (e.g. "tel:" or "ppn:").

The OMM will reply positively (i.e. no error) to this request if the delete request could be executed or queued.

The reply to *DeleteMessage* is an object called *DeleteMessageResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

Possible values for *errCode*:

Value	Description
ENoEnt	No DECT phone record found for given destination, this may also happen if the OMM does not know how to handle the URI scheme.
EInval	A field contains invalid data or exceeds a limit. The response contains the bad field.
ENoMem	The queue for this DECT phone is full. The client may try again after getting an <i>EventMessageQueueEmpty</i> for this DECT phone.

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Messaging*

4.55.5 CancelMessage

A client can send this request to remove a message from the send queue. If the message which has to be cancelled is still buffered in the OMM, it is removed from the queue. If it has been delivered already, this request has no effect.

Following fields are defined for the request *CancelMessage* in addition to the common attributes:

Name	Type	Mandatory	Description
sendTime	unsigned integer	yes	Original send time of the message to be cancelled.
id	unsigned integer	yes	Message ID.
toAddr	string	yes	Recipient address of the message to be cancelled. Must have the same scheme as the original message (e. g. "tel:" or

			"ppn:").
--	--	--	----------

The response will indicate success (i.e. no error code) if the request has been cancelled or if the queue was found but did not contain a fitting message anymore.

The reply to *CancelMessage* is an object called *CancelMessageResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

Possible values for *errCode*:

Value	Description
ENoEnt	No DECT phone record found for given destination, this may also happen if the OMM does not know how to handle the URI scheme.
EInval	A field contains invalid data or exceeds a limit. The response contains the bad field.
EFailed	The message to be cancelled was not found in the queue. The cause may be that it has been overridden by a higher priority message a short time ago or that it has been delivered to the DECT phone already.

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Messaging*

4.55.6 EventMessageQueueEmpty

This event is sent by OM AXI when the message queue of a DECT phone gets empty. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *MessageQueueEmpty*. This is allowed if the client has at least one of the following permissions: *Messaging* or *InfoMessaging*

Attributes sent in this notification:

Name	Type	Mandatory	Description
ppn	integer	yes	The DECT phone which's queue got empty.

Available filters: *ppn*

4.55.7 EventMessageProgress

This event is sent by OM AXI when the message queue of a DECT phone experiences some kind of progress. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *MessageProgress*. This is allowed if the client has at least one of the following permissions: *Messaging* or *InfoMessaging*

Attributes sent in this notification:

Name	Type	Mandatory	Description
------	------	-----------	-------------

sendTime	unsigned integer	yes	Original send time of the message.
id	unsigned integer	yes	Message ID.
toAddr	string	yes	Original recipient address of the message.
event	MessageProgressType	yes	The kind of progress.

is an enumeration type and can have one of these values:

Value	Description
Overridden	The message to be send has been removed from the queue by a higher prioritized message. If the client is still interested in sending this message, it should try again after getting an <i>EventQueueEmpty</i> .
DeleteOverridden	The message deletion commands have been removed from the queue by a higher prioritized message. If the client is still interested in deleting this message, it should try again after getting an <i>EventQueueEmpty</i> .
Delivered	The message has been delivered to the DECT phone. This does not mean that the user actually did open, read nor understand this message.
DeleteDelivered	This event is sent in one of these two cases: <ul style="list-style-type: none"> • The message deletion command has been delivered to the DECT phone • The message which should be deleted has been removed from the queue in case it has not been send to the DECT phone yet. In both cases the deletion can be seen as completed.
PagingTimeout	The DECT phone was not been found yet. However, the OMM will continue to search for this DECT phone. This is for informational purposes only. When <i>PagingTimeout</i> is received during the client is waiting for a <i>DeleteDelivered</i> the client must know, that the <i>DeleteMessage</i> is trashed. The <i>DeleteMessage</i> has to be sent again by the client after timeout.
Rejected	The message has not been accepted by the DECT phone. It should not be resent.
Busy	The message has not been accepted by the DECT phone. The DECT phone is temporarily busy and the message can be resent after a timeout.
WrongCharacterSet	The message has not been accepted by the DECT phone, because of a wrong character set. This must be the Latin1 character set.
WrongURI	The message has not been accepted by the DECT phone, because of an unknown URI format.
TemporaryUnavailable	The message has not been accepted by the DECT phone, because the DECT phone is switched off.
Unknown	The message has not been accepted by the DECT phone. The reason is unknown.

4.55.8 EventMessageConfirmation

This event is sent by the OM AXI when a notification session containing the type *MessageConfirmation* is active for a DECT phone and when there is a confirmation for a message of this DECT phone. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *MessageConfirmation*. This is allowed if the client has at least one of the following permissions: *Messaging* or *InfoMessaging*

Attributes sent in this notification:

Name	Type	Mandatory	Description
sendTime	unsigned integer	yes	Original send time.
id	unsigned integer	yes	Original message ID.
fromAddr	string	yes	Sender address of confirmed message. This is the destination of the confirmation.
toAddr	string	yes	Recipient address of confirmed message.
confTime	unsigned integer	yes	Time of confirmation, same format as sendTime.
confirmation	MessageConfirmType	yes	Confirmation for this message.

4.55.9 EventMessageSend

This event is sent by the OM AXI when a notification session containing the type *MessageSend* is active for a DECT phone or scheme and when there is a message sent **from** this DECT phone or **to** this address schema. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *MessageSend*. This is allowed if the client has at least one of the following permissions: *Messaging* or *InfoMessaging*

There is one element in this notification:

Name	Type	Mandatory	Description
alwaysIndirect	boolean	no	"1" or "true" when this message has to be routed by a messaging application. Note: This parameter is reserved for Mitel SIP-DECT internal use and certain behavior. So, it has not to be interpreted by other clients.
msg	MessageType	yes	Message to be sent.

Available filters: *ppn*, *scheme*

4.56 Integrated Message And Alarm Server Application Configuration And Maintenance

The integrated message and alarm server application (IMA) can be started and stopped using the request in the subsequent chapters.

4.56.1 GetIMA

With this request the client can query the configuration of IMA.

Following fields are defined for the request *Get/IMA* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *Get/IMAResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, if internal message routing from DECT phone to DECT phone is supported by IMA. “0” or “false”, if IMA supports only external message routing.
url	URLType	yes	URL of IMA configuration file.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.56.2 Set/IMA

With this request the client can set the configuration of IMA.

Following fields are defined for the request *Set/IMA* in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, if internal message routing from DECT phone to DECT phone is supported by IMA. “0” or “false”, if IMA supports only external message routing.
url	URLType	yes	URL of IMA configuration file.

The reply is an object called *Set/IMAResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.56.3 Event/IMACnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	no	The configuration for internal message routing from DECT phone to DECT phone changes.
url	URLType	yes	URL of IMA configuration file.

Available filters: (none)

4.57 OMP Java Web Start Configuration And Maintenance

The OMP Java Web start configuration using the request in the subsequent chapters.

4.57.1 GetOMPURL

With this request the client can query the OMP Java Web start configuration settings.

Following fields are defined for the request *GetOMPURL* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetOMPResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	URL settings for OMP Java Web start code base.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.57.2 SetOMPURL

With this request the client can set the OMP Java Web start configuration settings.

Following fields are defined for the request *SetOMPURL* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	URL settings for OMP Java Web start code base.

The reply is an object called *SetOMPURLResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.57.3 EventOMPURLCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	yes	URL settings for OMP Java Web start code base.

Available filters: (none)

4.58 Locating Configuration And Maintenance

4.58.1 LocationType

This type contains all data fields which describe the location of a DECT phone.

Name	Type	Description
id	integer	ID of RFP the DECT phone has been located at
rssiNew	integer	RSSI of the new RFP at the time of the handover in call state or for the next call setup in negative form, e.g. "-75" for -75 dBm
rssiOld	integer	RSSI of the old RFP at the time of the handover in call state or for the next call setup in negative form, e. g. "-75" for -75 dBm
nMultiFrame	unsigned integer	Multiframe number the location has been recorded at, (1 increment per 160 ms)

4.58.2 VisibilityType

This type contains all data fields which describe the visibility of an RFP to a DECT phone.

Name	Type	Description
id	integer	ID of RFP which can be seen by the DECT phone
rssiAvg	integer	Filtered RSSI value in negative form, e.g. "-75" for -75 dBm

4.58.3 RequestPositionInfo

With this request a client can locate a DECT phone.

Consider the license restrictions for this command (see 5.5).

Following fields are defined for the request *RequestPositionInfo* in addition to the common attributes:

Name	Type	Mandatory	Description
ppn	integer	yes	PPN of DECT phone to be found

The reply to *RequestPositionInfo* is an object called *RequestPositionInfoResp*. It does not contain information about the DECT phone location but only a confirmation if the Request was accepted or not. To actually receive a notification about the DECT phones location the client must be able to receive *EventPositionHistory* and *EventPositionInfo*, refer to the subsequent chapters.

RequestPositionInfoResp contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

Possible values for *errCode*:

Value	Description
ENoEnt	No record found for given ppn

EFailed	DECT phone is not in the right state to be located
EForbidden	It is not allowed to locate this DECT phone
EInProgress	This DECT phone still has a RequestPositionInfo to complete, try again later

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Locating*

4.58.4 SetPPUserTracking

The client can send this request to OM AXI to change the attribute *trackingActive* of a DECT phone user data set. The User ID (*uid*) has to be filled in by the client to identify the record to be changed. Consider the license restrictions for this command (see 5.5).

Following element is defined for the request *SetPPUserTracking* in addition to the common attributes:

Name	Type	Mandatory	Description
uid	integer	yes	User ID, numbering starts at 1, 0 is invalid
trackingActive	boolean	yes	„1” or “true”, if the location of this user has to be tracked

Note that tracking information is only sent to the client if it also has subscribed the event *EventPositionTrack* (4.58.7).

If the change fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetPPUserTrackingResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

Value	Description
EFailed	This user cannot be tracked
EForbidden	It is not allowed to locate or track this DECT phone user

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Locating, AllCnfWrite*

4.58.5 EventPositionHistory

This event is sent by OM AXI when information about the DECT phone position history has arrived. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Locating*. This is allowed if the client has at least one of the following permissions: *Locating*.

When a client wants to find out the current position and the position history of a DECT phone, it can send a *RequestPositionInfo*, as described in chapter 4.58.3. Since this information can only be found out asynchronously, it is sent in notifications of type *EventPositionHistory* to all clients who have an active notification.

This event contains following fields:

Name	Type	Mandatory	Description
ppn	integer	yes	PPN being located
nMultiFrame	unsigned integer	yes	Current multiframe number (1 increment per 160 ms)
loc	sequence of LocationType	yes	A list of the last know locations, the first one is the latest

Available filters: ppn

4.58.6 EventPositionInfo

This event is sent by OM AXI when information about the current DECT phone position has arrived. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Locating*. This is allowed if the client has at least one of the following permissions: *Locating*.

When a client wants to find out the current position and the position history of a DECT phone, it can send a *RequestPositionInfo*, as described in chapter 4.58.3. Since this information can only be found out asynchronously, it is sent in notifications of type *EventPositionInfo* to all clients who have an active notification.

When the DECT phone did not respond, this Event is sent with an empty *vis* list. If not received already, this also means implicitly that *EventPositionHistory* will not arrive.

This event contains following fields:

Name	Type	Mandatory	Description
ppn	integer	yes	PPN being located
vis	sequence of VisibilityType	yes	A list of all RFPs currently visible

Available filters: *ppn*

4.58.7 EventPositionTrack

This event is sent by OM AXI when DECT phone position tracking information has arrived. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *PositionTrack* for one or more DECT phones. This is allowed if the client has at least one of the following permissions: *Locating*.

Each time one of the DECT phones which are linked with a user having the attribute *trackingActive* set locks to another RFP, an *EventPositionTrack* is sent by OM AXI.

Note that tracking a DECT phone will increase its power consumption.

This event contains following fields:

Name	Type	Mandatory	Description
ppn	integer	yes	PPN being tracked
id	integer	yes	ID of current RFP

Available filters: *ppn*

4.58.8 EventPositionRequest

This event is sent by OM AXI when a user wants to locate another DECT phone using his DECT phone.

To get this event the client has to subscribe to *Locating*. This is allowed if the client has at least one of the following permissions: *Locating*.

When a client wants to find out the current position and the position history of a DECT phone, it can send a *RequestPositionInfo*, as described in chapter 4.58.3.

This event contains following fields:

Name	Type	Mandatory	Description
num	string	yes	Number of user who wants to locate somebody
name	string	no	Name of the user who wants to locate somebody, optional and for informational purposes only
targetNum	string	yes	Number of user who has to be located
targetName	string	no	Name of the user who has to be located, optional and for informational purposes only.

Available filters: *ppn*

The filter *ppn* applies for the DECT phone of the locating user.

4.59 User Monitoring Configuration And Maintenance

4.59.1 GetUserMonitoring

This request is sent from the client to the OMM.

With this request the client can query common information about the user monitoring configuration.

Following fields are defined for the request *GetUserMonitoring* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetUserMonitoring* is an object called *GetUserMonitoringResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
startupDelay	integer	yes	Period of time in seconds after OMM start up before any unavailability is checked and escalated. Range: 120-900 sec / means 2-15 min
escalationDelay	integer	yes	Period of time in seconds before the unavailability is escalated. Range: 0-300 sec / means 0-5 min
activityTimeout1	integer	yes	Period of time a DECT phone was inactive before the DECT phone activity status is set to "unavailable". Range: 1800-86400

			sec / means 30min-24 h. This attribute is used for passive user monitoring (see PPUserType in chapter 4.39.1).
activityTimeout2	integer	yes	Time interval of periodic DECT activities. DECT phone activity timeout before the OMM initiates an activation between the DECT phone and the DECT system. Range: 300-3600 sec / means 5-60 min. This attribute is used for active user monitoring (see PPUserType in chapter 4.39.1).
locatingEscalation	boolean	yes	Enables sending the alarm trigger "LOC-ERROR-USERSTATE".
batteryThresholdValue	integer	yes	Battery level threshold value (0-100%) to escalate low battery power for a DECT phone.

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.59.2 SetUserMonitoring

This request is sent from the client to the OMM.

With this request the client can set the configuration of the user monitoring.

Following fields are defined for the request *SetUserMonitoring* in addition to the common attributes:

Name	Type	Mandatory	Description
startupDelay	integer	no	Period of time in seconds after OMM start up before any unavailability is checked and escalated. Range: 120-900 / means 2-15 min
escalationDelay	integer	no	Period of time in seconds before the unavailability is escalated. Range: 0-300 / means 0-5 min
activityTimeout1	integer	no	Period of time a DECT phone was inactive before the DECT phone activity status is set to "unavailable". Range: 1800-86400 sec / means 30min-24 h. This attribute is used for passive user monitoring (see PPUserType in chapter 4.39.1).
activityTimeout2	integer	no	Time interval of periodic DECT activities. DECT phone activity timeout before the OMM initiates an activation between the DECT phone and the DECT system. Range: 300-3600 sec / means 5-60 min. This attribute is used for active user monitoring (see PPUserType in chapter 4.39.1).
locatingEscalation	boolean	no	Enables sending the alarm trigger "LOC-ERROR-USERSTATE".
batteryThresholdValue	integer	no	Battery level threshold value (0-100%) to escalate low battery power for a DECT phone.

The reply to *SetUserMonitoring* is an object called *SetUserMonitoringResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.59.3 EventUserMonitoringCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
startupDelay	integer	no	Period of time in seconds after OMM start up before any unavailability is checked and escalated. Range: 120-900 / means 2-15 min
escalationDelay	integer	no	Period of time in seconds before the unavailability is escalated. Range: 0-300 / means 0-5 min
activityTimeout1	integer	yes	Period of time a DECT phone was inactive before the DECT phone activity status is set to "unavailable". Range: 1800-86400 sec / means 30min-24 h. This attribute is used for passive user monitoring (see PPUserType in chapter 4.39.1).
activityTimeout2	integer	yes	Time interval of periodic DECT activities. DECT phone activity timeout before the OMM initiates an activation between the DECT phone and the DECT system. Range: 300-3600 sec / means 5-60 min. This attribute is used for active user monitoring (see PPUserType in chapter 4.39.1).
locatingEscalation	boolean	no	Enables sending the alarm trigger "LOC-ERROR-USERSTATE".
batteryThresholdValue	integer	no	Battery level threshold value (0-100%) to escalate low battery power for a DECT phone.

Available filters: (none)

4.60 XML Application Configuration And Maintenance

4.60.1 XMLApplicationType

This type contains all data fields of an XML application setting. It is used in different requests and responses defined in this document.

The *XMLApplicationType* is structured follows:

Name	Type	Description
------	------	-------------

enable	boolean	„1” or “true”, the XML application shall be activated
id	integer	ID of the XML application. This value cannot be used by the client in <i>CreateXMLApplication</i> .
name	string	<p>Name of the XML application For built-in XML applications predefined names are used. The following built-in names for XML applications are defined:</p> <p>callerList External PBX caller list support The local DECT phone caller list is disabled when set.</p> <p>redialList External PBX redial list support The local DECT phone redial list is disabled when set.</p> <p>userPresence XML user presence application support by an external server.</p> <p>systemAppMenu XML system application menu support by an external server.</p> <p>featureAccessCodes XML application to translate feature access codes.</p> <p>callCompletion Call completion URI support to notify call completion to an external XML server.</p> <p>parkCall XML application hook URI supporting call park in active state by an external server.</p> <p>unparkCall XML application hook URI supporting call unpark in idle state by an external server.</p> <p>pickup XML application hook URI supporting pickup in idle state by an external server.</p> <p>take XML application hook URI supporting take in idle state by an external server.</p> <p>callForward XML application hook URI supporting call forward in idle state by an external server.</p> <p>callRouting XML application hook URI supporting call routing in idle state by an external server.</p> <p>callProtection XML application hook URI supporting call protection in idle state by an external server.</p> <p>voiceBox XML application hook URI supporting call voice box in idle state by an external server.</p> <p>admin OMM internal XML administration application provided in special OMM systems. This application is read only and can only be used as information. It provides the built in possibility to administrate parts of the OMM configuration data using a DECT phone.</p> <p>eventActions Call event action URI support to an external XML server.</p>
type	string	Type of the XML application. One of “BuiltIn”, “BuiltInReadOnly”, “Dynamic” or “CorpDir”. Note, that the type “BuiltIn” cannot be set.
corpDirOrder	integer	Order number to prioritize this XML directory data set. This value is only used for XML type “CorpDir”

		<p>This setting might affect the order number in other corporate directory data sets. A reordering of all corporate directory data sets will be performed. If empty, the XML directory data set is queued as last entry of all corporate directory data sets.</p> <p>See also at the <i>order</i> number of the LDAP directories in 4.52.1.1.</p>
url	URLType	<p>User data server configuration settings</p> <p>Note: When the XML server is located on the call server the host in the “URLType” can be addressed by “SIPProxy” placeholder. In this case the user depended call server is used where the user is registered. The path of the <i>URLType</i> includes the path, query and fragment of the URI, e. g. “<i>omm.callLists?key=17&cnt={count}&na={number}</i>” As shown the URLType path contains predefined replacements. The following predefined replacements are known by the OMM: {<i>count</i>}: Number of items to be requested from the XML server {<i>subsc</i>}or {<i>number</i>}: Identification of the XML client e. g. its subscriber number {<i>sicha</i>}: Silent charging indication {<i>boot</i>}: End of boot sequence indication {<i>reg</i>}: Successful registration indication {<i>onho</i>}: On-hook indication {<i>offho</i>}: Off-hook indication {<i>in</i>}: Incoming call indication {<i>out</i>}: Outgoing call indication {<i>poll</i>}: Time based indication {<i>sip</i>}: SIP Notify indication {<i>con</i>}: Connect indication {<i>dis</i>}: Disconnect indication {<i>rege</i>}: Registration event indication {<i>sipsrc</i>}: SIP source address {<i>sipdest</i>}: SIP destination address</p>

4.60.2 CreateXMLApplication

The client can send this request to OM AXI to create a new XML application setting. Built-in application settings cannot be created.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
xmlAppl	XMLApplicationType	yes	Data of XML application to create

If the creation was successful, the reply contains all attributes which *GetXMLApplication* would return. Some data fields may be changed or set by the OMM. If the creation fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreateXMLApplicationResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.60.3 DeleteXMLApplication

With this request the client can delete an XML application setting in the OMM. The key is the XML application set *id*. Built-in XML applications cannot be deleted.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	ID of the XML application to be deleted

The reply to this request is an object called *DeleteXMLApplicationResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description

Possible values for *errCode*:

Value	Description
ENoEnt	XML application for the given <i>id</i> not found

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.60.4 GetXMLApplication

With this request the client can acquire one or all XML application settings from the OMM. The request contains an optional XML application *id* to be fetched. If no *id* is given, all XML application settings will be fetched.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	no	ID of the XML application to be got.

The reply to *GetXMLApplication* is an object called *GetXMLApplicationResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
xmlAppl	Sequence of XMLApplicationType	yes	Data of XML application to be got.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.60.5 SetXMLApplication

The client can send this request to OM AXI to change an XML application setting. The *id* has to be filled in by the client to identify the XML application to be changed. Additionally the attributes which have to be changed must be filled in by the client.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
xmlAppl	XMLApplicationType	yes	Data of record to be changed

If the change was successful, the reply contains all attributes which *GetXMLApplication* would return. If the change fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetXMLApplicationResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.60.6 EventXMLApplicationCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains the XML application setting which have been changed.

Name	Type	Mandatory	Description
xmlAppl	XMLApplicationType	yes	Changed data of XML application

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with only an *id* of the deleted *xmlAppl* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
id	integer	yes	ID of the deleted XML application
name	string	yes	Name of the deleted XML application
deleted	boolean, default false		"1", the XML application data setdata set has been deleted

Available filters: (none)

4.61 Conference Server Configuration And Maintenance

4.61.1 ConferenceRoomType

This type contains all data fields of a conference room setting. It is used in different requests and responses defined in this document.

The *ConferenceRoomType* is structured as follows:

Name	Type	Description
id	integer	Internal <i>id</i> of the conference room. This value can be left by the client in <i>CreateConferenceRoom</i> .

name	string	Name of the conference room
conferenceId	string	Conference identification (conf-id / number or URI)
sipAuthId	string	SIP authentication of the conference room user
sipPw	string	SIP authentication password of the conference room user, encrypted with public key
fixedSipPort	integer	Explicitly used SIP port for registering the conference room at the call server. If <i>fixedSipPort</i> =0, no specific port is used for this conference room and the <i>calculatedSipPort</i> is used.
calculatedSipPort	integer	Calculated SIP port for registering the conference room at the call server. If <i>calculatedSipPort</i> =0, the <i>fixedSipPort</i> is used. This value is read only.

4.61.2 ConferenceCodecType

This type contains all data fields of a conference codec. It is used in different requests and responses defined in this document.

The *ConferenceCodecType* is structured follows:

Name	Type	Description
mediaCodec	MediaCodecType	Media codec identification
mediaPt	integer	Media payload type/value identification. Payload values are restricted to 7 bits hence 128 is invalid PT_G711_U = 0, PT_G723_53 = 4, PT_G723_63 = PT_G723_53, PT_G711_A = 8, PT_G722 = 9, PT_G729 = 18, PT_G729A = PT_G729, PT_G729B = PT_G729, PT_G729AB = PT_G729, PT_H261 = 31, PT_H263 = 34, PT_DYN_FIRST = 96, PT_DYN_LAST = 127, PT_NOT_SUPPORTED = 128
rate	integer	Sample rate in kHz, currently 8 or 16 for audio. For video: currently w/o meaning, must be set to 0.

MediaCodecType can have one of the following values:

Value	Description
CodecG711_A	G711 a-law codec
CodecG723_53	G.723 5.3 kb codec

CodecG723_63	G.723 6.3 kb codec
CodecG729	G.729 codec
CodecG711_U	G711 μ -law codec
CodecG722	G.722 codec
CodecH261	H.261 video codec
CodecH263	H.263 video codec
CodecG726_16	G.726 16 kb codec (little endian variant)
CodecG726_24	G.726 24 kb codec (little endian variant)
CodecG726_32	G.726 32 kb codec (little endian variant)
CodecG726_40	G.726 40 kb codec (little endian variant)
CodecAAL2_G726_16	G.726 16 kb codec (big endian variant)
CodecAAL2_G726_24	G.726 24 kb codec (big endian variant)
CodecAAL2_G726_32	G.726 32 kb codec (big endian variant)
CodecAAL2_G726_40	G.726 40 kb codec (big endian variant)
CodecL16	Dynamic L16 family codec
CodecH264	Dynamic H.264 video codec
CodecInvalid	None or invalid codec

4.61.3 ConferenceDataType

This type contains all data fields of a conference codec. It is used in different requests and responses defined in this document.

The *ConferenceDataType* is structured follows:

Name	Type	Mandatory	Description
mixerId	integer	yes	Identification of the conference mixer
num	string	yes	Number of the added user. This value mandatory but only necessary for monitoring aspects. It does not influence the conference handling.
localRTP	integer	yes	Local RTP port number
localRTCP	integer	yes/no	Local RTCP port number; this value is optional in <i>ChangeUserInConference</i> and will be ignored.

remoteRTP	integer	yes	Remote RTP port number
remoteRTCP	integer	yes	Remote RTCP port number
remotelpAddr	string	yes	IP address of the remote party
vif	integer	yes	“Voice in Frame” / use packet size
vad	boolean	yes	„1” or “true”, if “voice activity detection (silence suppression) is enabled
txCodeclDx	integer	yes	Tx codec index to the codec list below
rxCodeclDx	integer	yes	Rx codec index to the codec list below
codec	Sequence of ConferenceCodecType	yes	List of usable codecs.
srtpProfile	string	no	SRTP profile type. One of "None", "AES128_CM_SHA1_32", "AES128_CM_SHA1_80", "NULL_SHA1_32" or "NULL_SHA1_80". Default setting is “None”
txSecret	string	no	SRTP Tx stream secret.
rxSecret	string	no	SRTP Rx stream secret.

4.61.4 CreateConferenceRoom

The client can send this request to OM AXI to create a new conference room setting.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
room	ConferenceRoomType	yes	Data of conference room settings to create

If the creation was successful, the reply contains all attributes which *GetConferenceRoom* would return. Some data fields may be changed or set by the OMM. If the creation fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreateConferenceRoomResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.61.5 DeleteConferenceRoom

With this request the client can delete an conference room setting in the OMM. The key is the conference room set *id*.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

id	integer	yes	Internal <i>id</i> of the conference room to be deleted
----	---------	-----	---

The reply to this request is an object called *DeleteConferenceRoomResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description

Possible values for *errCode*:

Value	Description
ENoEnt	conference room for the given <i>id</i> not found

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.61.6 GetConferenceRoom

With this request the client can acquire one or more conference room settings from the OMM. The request contains an *id* from which starting subsequent conference room data setdata sets are to be fetched. The client can chose the number of records returned. If less than this number of data setdata sets are returned or *ENoEnt*, the client knows that it reached end of file.

id s usually ordered start at 0, but possibly not all numbers are assigned, i.e. the conference room numbering may not be contiguous.

The responses are by *id*, starting with the smallest.

If the client asked for a certain number records but the given *id* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given *id* has to match to a record exactly. If there is a record with this *id*, it is returned, otherwise *ENoEnt*.

To acquire all data setdata sets, one could send *GetConference* (*id* = "0", *maxRecords* = "20"). If the response contains 20 records, take the last *id* plus one and send another *GetConference*. Repeat this until you get less than 20 records back.

Following fields are defined for the request *GetConferenceRoom* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	Internal <i>id</i> of the conference room to be got
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to *GetConferenceRoom* is an object called *GetConferenceRoomResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
room	Sequence of ConferenceRoomType	no	Data conference room to be got.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.61.7 SetConferenceRoom

The client can send this request to OM AXI to change a conference room setting. The *id* has to be filled in by the client to identify the conference room to be changed. Additionally the attributes which have to be changed must be filled in by the client.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
room	ConferenceRoomType	yes	Data of conference room to be changed.

If the change was successful, the reply contains all attributes which *GetConferenceRoom* would return. If the change fails, OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetConferenceRoomResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.61.8 EventConferenceRoomCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *ConferenceCnf*. This is allowed if the client has at least one of the following permissions: *AllReadCnf*, *Conferencing* or *Monitoring*.

The notification contains the conference room setting which have been changed.

Name	Type	Mandatory	Description
room	ConferenceRoomType	yes	Changed data of conference room
deleted	boolean, default false	yes	„1“, the conference room data setdata set has been deleted

This event is also sent upon creation of a record. If the record has been deleted, this event is sent with only an *id* of the deleted *ConferenceRoomType* element and the attribute *deleted="1"*.

Available filters: (none)

4.61.9 CreateConference

The client can send this request to OM AXI to create a new conference for a conference server.

Following element is defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
conferenceId	string	yes	Conference identification (conf-id / number or URI). This value mandatory but only necessary for monitoring aspects. It does not influence the conference handling.

The reply to this request is called *CreateConferenceResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
mixerId	integer	yes	Identification of the conference mixer
ipAddr	string	yes	IP address of the conference mixer
port	integer	yes	RTP port of the first subscriber

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *Conferencing*

4.61.10 DeleteConference

With this request the conference server client can delete a conference in the OMM. The key is the conference *id*.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
mixerId	integer	yes	Identification of the conference mixer

The reply to this request is an object called *DeleteConferenceResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *Conferencing*

4.61.11 AddUserToConference

With this request the conference server client can add a subscriber user to an existing conference.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
conference	ConferenceDataType	yes	Conference data of user to add.

The reply to *AddUserToConference* is an object called *AddUserToConferenceResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *Conferencing*

4.61.12 DeleteUserFromConference

With this request the conference server client can delete a subscriber user from an existing conference.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
mixerId	integer	yes	Identification of the conference mixer
localRTP	integer	yes	Local RTP port number

The reply to *DeleteUserFromConference* is an object called *DeleteUserFromConferenceResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *Conferencing*

4.61.13 ChangeUserInConference

With this request the conference server client can change a subscriber user in an existing conference.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
conference	ConferenceDataType	yes	Conference data of user to change.

The reply to *ChangeUserInConference* is an object called *ChangeUserInConferenceResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *Conferencing*

4.61.14 SetRTPConferenceStreamChg

With this request the conference server client can restart and stop the RTP streams an existing conference.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description															
mixerId	integer	yes	Identification of the conference mixer															
localRTP	integer	yes	Local RTP port number															
rx	boolean	yes	<p>“1” or “true”, if the conference mixer RX stream shall be used.</p> <table> <thead> <tr> <th>Old state</th> <th>Setting</th> <th>New state</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>1</td> <td>RX restart</td> </tr> <tr> <td>0</td> <td>0</td> <td>RX unchanged</td> </tr> <tr> <td>1</td> <td>1</td> <td>RX unchanged</td> </tr> <tr> <td>1</td> <td>0</td> <td>RX stopped</td> </tr> </tbody> </table>	Old state	Setting	New state	0	1	RX restart	0	0	RX unchanged	1	1	RX unchanged	1	0	RX stopped
Old state	Setting	New state																
0	1	RX restart																
0	0	RX unchanged																
1	1	RX unchanged																
1	0	RX stopped																

tx	boolean	yes	“1” or “true”, if the conference mixer TX stream shall be used. See also RX above for TX
----	---------	-----	--

The reply to *ChangeUserInConference* is an object called *ChangeUserInConferenceResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Conferencing*

4.61.15 EventCreateConference

This event is sent by OM AXI when a conference has been created. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Conference*. This is allowed if the client has at least one of the following permissions: *Conferencing* or *Monitoring*.

The notification contains the conference setting which has been created.

Name	Type	Mandatory	Description
mixerId	integer	yes	Identification of the conference mixer.
conferenceId	string	yes	Conference identification (conf-id / number or URI).
rfpId	integer	yes	RFP id of this conference
ipAddr	string	yes	IP address of the conference server
port	integer	yes	RTP port of the first subscriber

Available filters: (none)

4.61.16 EventDeleteConference

This event is sent by OM AXI when a conference has been deleted. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Conference*. This is allowed if the client has at least one of the following permissions: *Conferencing* or *Monitoring*.

The notification contains the conference setting which has been deleted.

Name	Type	Mandatory	Description
mixerId	integer	yes	Identification of the conference mixer.
conferenceId	string	yes	Conference identification (conf-id / number or URI)

Available filters: (none)

4.61.17 EventAddUserToConference

This event is sent by OM AXI when a user has been added to a conference. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Conference*. This is allowed if the client has at least one of the following permissions: *Conferencing* or *Monitoring*.

The notification contains the setting of the user which has been added to the conference.

Name	Type	Mandatory	Description
conference	ConferenceDataType	yes	Conference data of user added.

Available filters: (none)

4.61.18 EventChangeUserInConference

This event is sent by OM AXI when a user has been added to a conference. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Conference*. This is allowed if the client has at least one of the following permissions: *Conferencing* or *Monitoring*.

The notification contains the setting of the user which has been added to the conference.

Name	Type	Mandatory	Description
conference	ConferenceDataType	yes	Conference data of user changed.

Available filters: (none)

4.61.19 EventDeleteUserFromConference

This event is sent by OM AXI when a user has been added to a conference. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Conference*. This is allowed if the client has at least one of the following permissions: *Conferencing* or *Monitoring*.

The notification contains the setting of the user which has been added to the conference.

Name	Type	Mandatory	Description
mixerId	integer	yes	Identification of the conference mixer.
num	string	yes	Number of the deleted user.
localRTP	integer	yes	Local RTP port number

Available filters: (none)

4.61.20 EventRTPConferenceStreamChg

This event is sent by OM AXI when a RTP streams changes within a conference. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Conference*. This is allowed if the client has at least one of the following permissions: *Conferencing* or *Monitoring*.

The notification contains the setting of the user which has been added within a conference.

Name	Type	Mandatory	Description
mixerId	integer	yes	Identification of the conference mixer.
num	string	yes	Number of the added user.
localRTP	integer	yes	Local RTP port number
rx	boolean	yes	"1" or "true", if the conference mixer RX stream is used.
tx	boolean	yes	"1" or "true", if the conference mixer TX stream is used.

Available filters: (none)

4.61.21 EventConferenceRequest

This event is sent by OM AXI when a conference request has been initiated by a DECT phone. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Conference*. This is allowed if the client has at least one of the following permissions: *Conferencing* or *Monitoring*.

The notification contains the conference request parameters.

Name	Type	Mandatory	Description
sipRef	string	yes	Reference to the SIP instance that requests the conference room.
confUsers	integer	yes	Maximal number of conference users.

Available filters: (none)

4.61.22 EventConferenceRelease

This event is sent by OM AXI when a conference call to a conference room is released or failed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Conference*. This is allowed if the client has at least one of the following permissions: *Conferencing* or *Monitoring*.

The notification contains the conference request parameters.

Name	Type	Mandatory	Description
------	------	-----------	-------------

sipRef	string	yes	Reference to the SIP instance that requests the conference room.
conferenceId	string	yes	Conference identification (conf-id / number or URI)

Available filters: (none)

4.61.23 ConferenceConfirmation

With this request the conference server can confirm a prior initiated conference by a DECT phone (see 4.61.21).

This request is sent by the conference server when the request for a conference has been successfully performed.

Name	Type	Mandatory	Description
sipRef	string	yes	Reference to the SIP instance that requests the conference room.
conferenceId	string	yes	Conference identification (conf-id / number or URI)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Conferencing*

The request response called *ConferenceConfirmationResp* contains no parameters.

Name	Type	Mandatory	Description

4.61.24 GetG729ChannelsForConference

With this request the conference server client can request a number of G.729 channels for a conference. The response contains the absolute number of G.729 channels the conference server has to use.

Note: This request is only for internal use by the Integrated Conference Server (ICS)

This request is sent by the client and contains the following parameters.

Name	Type	Mandatory	Description
channels	integer	yes	Number of channels the conference server wants to use.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Conferencing*

The request response called *GetG729ChannelsForConferenceResp* contains the following parameters.

Name	Type	Mandatory	Description
channels	integer	yes	Number of channels the conference server shall use.

4.61.25 GetReadyForConferencing

With this request the conference server client can request if all OMM SIP accounts are ready for conferencing.

This request is sent by the client and has no parameters.

Name	Type	Mandatory	Description

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Conferencing*

The request response called *GetReadyForConferencingResp* contains the following parameters.

Name	Type	Mandatory	Description
ready	boolean	yes	„1” or “true”, if “all OMM SIP accounts are ready for conferencing.

4.61.26 EventReadyForConferencing

This event is sent by OM AXI when all OMM SIP accounts are ready for conferencing. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Conference*. This is allowed if the client has at least one of the following permissions: *Conferencing* or *Monitoring*.

The notification contains the conference room parameters.

Name	Type	Mandatory	Description

Available filters: (none)

4.61.27 GetFreeConferenceChannels

With this request the conference server client can request the number of free conference channels.

This request is sent by the client and has no parameters.

Name	Type	Mandatory	Description

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Conferencing*

The request response called *GetFreeConferenceChannelsResp* contains the following parameters.

Name	Type	Mandatory	Description
freeChannels	integer	yes	Number of free conference channels.
totalChannels	integer	yes	Number of total conference channels..

4.61.28 EventFreeConferenceChannels

This event is sent by OM AXI, when the number of free conference channels changes. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Conference*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains the conference room parameters.

Name	Type	Mandatory	Description
freeChannels	integer	yes	Number of free conference channels.
totalChannels	integer	yes	Number of total conference channels..

Available filters: (none)

4.62 Bluetooth Beacon Configuration And Maintenance

4.62.1 BTRfpSensType

This type contains all configuration data fields of global RFP Bluetooth beacon sensitivity level settings. It is used in different requests and responses defined in this document. Not all fields are used in all OMM versions and in all request and response types.

Name	Type	Description
checkpoint	int	Checkpoint threshold value level for Bluetooth beacons at RFPs. Possible value range is -1 (lowest) till -119 (highest). Default value setting for checkpoint is -35.
low	int	Low sensitivity threshold value level for Bluetooth beacons at RFPs. Possible value range is -1 (lowest) till -119 (highest). Default value setting for low is -55.
medium	int	Medium sensitivity threshold value level for Bluetooth beacons at RFPs. Possible value range is -1 (lowest) till -119 (highest). Default value setting for medium is -75.
high	int	High sensitivity threshold value level for Bluetooth beacons at RFPs. Possible value range is -1 (lowest) till -119 (highest). Default value setting for high is -100.

4.62.2 BTPpSensType

This type contains all configuration data fields of global DECT phone Bluetooth module sensitivity level settings. It is used in different requests and responses defined in this document. Not all fields are used in all OMM versions and in all request and response types.

Name	Type	Description
------	------	-------------

low	int	Low sensitivity level for Bluetooth modules at DECT phones. Possible value range is -1 (lowest) till -119 (highest). Default value setting for low is -55.
medium	int	Medium sensitivity level for Bluetooth modules at DECT phones. Possible value range is -1 (lowest) till -119 (highest). Default value setting for medium is -75.
high	int	High sensitivity level for Bluetooth modules at DECT phones. Possible value range is -1 (lowest) till -119 (highest). Default value setting for high is -100.

4.62.3 BTBType

This type contains all configuration data fields and state information about a Bluetooth beacon. It is used in different requests and responses defined in this document. Not all fields are used in all OMM versions and in all request and response types.

Name	Type	Description
id	integer	Unique Bluetooth beacon identifier. The numbering starts at 0
name	string	Unique Bluetooth beacon name identifier.
ethAddr	string	Ethernet address, format "00:11:22:aa:bb:cc"
rfpld	integer	RFP id the Bluetooth beacon is connected to
hierarchy1	string	Position hierarchy level 1, free format text
hierarchy2	string	Position hierarchy level 2, free format text
hierarchy3	string	Position hierarchy level 3, free format text
hierarchy4	string	Position hierarchy level 4, free format text
rssI	integer	Threshold to detect a Bluetooth beacon device (in dBm, e. g. – 75)
calibration	integer	Calibration of the Bluetooth beacon device
state	string	One of "unplugged", "plugged" or "failed". This element is read only and cannot be set.
checkpoint	boolean	„1" or "true", if "this Bluetooth beacon is enabled as checkpoint.

In a *CreateBluetoothBeacon* request the field *ethAddr* is optional. In a *SetBluetoothBeacon* request the field's *id* is mandatory. In all other case the fields are optional. The successes of the requests depend further on the permissions of the user being logged in.

4.62.4 GetBluetoothGlobalSettings

With this request the client can query information the global Bluetooth settings from the OMM.

Following fields are defined for the request *GetBluetoothGlobalSettings* in addition to the common attributes:

Name	Type	Mandatory	Description

The reply to *GetBluetoothGlobalSettings* is an object called *GetBluetoothGlobalSettingsResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
active	boolean	yes	„1” or “true”, if this the general DECT phone Bluetooth setting is enabled .

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.62.5 SetBluetoothGlobalSettings

The client can send this request to OM AXI to change global Bluetooth settings.

Following fields are defined for the request *SetBluetoothGlobalSettings* in addition to the common attributes:

Name	Type	Mandatory	Description
active	boolean	no	„1” or “true”, if this the general DECT phone Bluetooth setting is enabled .

If the global Bluetooth settings change was successful, the reply contains all Bluetooth settings attributes which *GetBluetoothGlobalSettings* would return. If the creation fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetBluetoothGlobalSettingsResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.62.6 EventBluetoothGlobalSettingsCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *BluetoothCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description

active	boolean	no	„1” or “true”, if this the general DECT phone Bluetooth setting is enabled .
--------	---------	----	--

Available filters: no

4.62.7 GetBluetoothSensitivity

With this request the client can query information the global Bluetooth sensitivity settings from the OMM.

Following fields are defined for the request *GetBluetoothSensitivity* in addition to the common attributes:

Name	Type	Mandatory	Description

The reply to *GetBluetoothSensitivity* is an object called *GetBluetoothSensitivityResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
rfpSens	BTRfpSensType	yes	Sensitivity level for Bluetooth beacons connected to RFPs.
ppSens	BTPpSensType	yes	Sensitivity level for Bluetooth modules at DECT phones.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.62.8 SetBluetoothSensitivity

The client can send this request to OM AXI to change global Bluetooth sensitivity settings.

Following fields are defined for the request *SetBluetoothSensitivity* in addition to the common attributes:

Name	Type	Mandatory	Description
rfpSens	BTRfpSensType	no	Sensitivity level for Bluetooth beacons connected to RFPs.
ppSens	BTPpSensType	no	Sensitivity level for Bluetooth modules at DECT phones.

If the global Bluetooth sensitivity change was successful, the reply contains all Bluetooth sensitivity attributes which *GetBluetoothSensitivity* would return. If the creation fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetBluetoothSensitivityResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.62.9 EventBluetoothSensitivityCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *BluetoothCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
rfpSens	BTRfpSensType	no	Sensitivity level for Bluetooth beacons connected to RFPs.
ppSens	BTPpSensType	no	Sensitivity level for Bluetooth modules at DECT phones.

Available filters: no

4.62.10 GetBluetoothBeacon

With this request the client can query information about one or more Bluetooth beacons from the OMM.

The request contains a Bluetooth beacon-ID from which starting subsequent data setdata sets are to be fetched. The client can chose the number of records returned. If less than this number of data setdata sets are returned or *ENoEnt*, the client knows that it reached end of file.

Bluetooth beacon-IDs usually start at 0, but possibly not all numbers are assigned, i.e. the data setdata set numbering may not be contiguous.

The responses are ordered by *id*, starting with the smallest.

If the client asked for a certain number records but the given *id* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given *id* has to match to a record exactly. If there is a record with this *id*, it is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetBluetoothBeacon* (*id* = "0", *maxRecords* = "20"). If the response contains 20 records, take the last *id* plus one and send another *GetBluetoothBeacon*. Repeat this until you get less than 20 records back.

Following fields are defined for the request *GetBluetoothBeacon* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	First Bluetooth beacon id of Bluetooth beacons to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to *GetBluetoothBeacon* is an object called *GetBluetoothBeaconResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
btb	sequence of BTBType	no	Bluetooth beacon record, if found

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.62.11 CreateBluetoothBeacon

The client can send this request to OM AXI to create a new Bluetooth beacon data set. beacon

If there is no *id* assigned by the client, the OMM will choose one. If the client assigns an *id*, it must not exist yet.

Following fields are defined for the request *CreateBluetoothBeacon* in addition to the common attributes:

Name	Type	Mandatory	Description
btb	BTBType	yes	Data of Bluetooth beacon to create.

If the Bluetooth beacon creation was successful, the reply contains all Bluetooth beacon attributes which *GetBluetoothBeacon* would return, but no state information. Some data fields may be changed or set by the OMM. If the creation fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreateBluetoothBeaconResp*. It contains all actual set attributes, all attributes are optional.

Possible values for *errCode*:

Value	Description
ENoMem	No more Bluetooth beacons can be created - the system is completely configured

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

Note: Bluetooth beacons can be hot plugged and are identified by their MAC addresses. Therefore it is not needed to create each Bluetooth beacon, but it can be done.

4.62.12 DeleteBluetoothBeacon

With this request the client can delete a Bluetooth beacon record in the OMM. The key used to delete a Bluetooth beacon is its *id*.

Following fields are defined for the request *DeleteBluetoothBeacon* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	id of Bluetooth beacon to delete

The reply to *DeleteBluetoothBeacon* is an object called *DeleteBluetoothBeaconResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.62.13 SetBluetoothBeacon

The client can send this request to OM AXI to change a Bluetooth beacon data set. The *id* has to be filled in by the client to identify the record to be changed. Additionally the attributes of the Bluetooth beacon which have to be changed must be filled in by the client.

Note that attributes describing transient properties are ignored.

Following fields are defined for the request *SetBluetoothBeacon* in addition to the common attributes:

Name	Type	Mandatory	Description
btb	BTBType	yes	Data of Bluetooth beacon to change.

If the Bluetooth beacon change was successful, the reply contains all Bluetooth beacon attributes which *GetBluetoothBeacon* would return, but no state information. Some data fields may be changed or set by the OMM. If the creation fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetBluetoothBeaconResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.62.14 EventBluetoothBeaconCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *BluetoothCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

This event is also sent upon creation of a Bluetooth beacon. If the Bluetooth beacon has been deleted, this event is sent with only an *rfpld* in the *rfp* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
btb	BTBType	yes	Contains all attributes which have been changed.
deleted	boolean, default false		„1” or “true”, if this data set has been deleted

Available filters: no

4.62.15 GetBluetoothBeaconSummary

With this request the client can query common information about the Bluetooth beacons configured.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to this request is an object called *GetBlueToothBeaconSummaryResp*. It contains following fields in addition to the common attributes:

Name	Type	Description
nRecords	integer	The total number of Bluetooth beacons set up.

idFirst	integer	The ID of the first Bluetooth beacon set up.
checkpointNum	integer	Number of Bluetooth beacons enabled as checkpoint.
stateUnpluggedNum	integer	Number of unplugged Bluetooth beacons.
statePluggedNum	integer	Number of plugged Bluetooth beacons.
stateFailedNum	integer	Number of failed Bluetooth beacons.

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.62.16 EventBluetoothBeaconSummary

This event is sent by OM AXI when the number of Bluetooth beacon summary counters have been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *BluetoothBeaconSumCnf* or to *BluetoothBeaconSumState*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Description
nRecords	integer	The total number of Bluetooth beacons set up (when subscribed to <i>BluetoothBeaconSumCnf</i>).
idFirst	integer	The ID of the first Bluetooth beacon set up (when subscribed to <i>BluetoothBeaconSumCnf</i>).
checkpointNum	integer	Number of Bluetooth beacons enabled as checkpoint (when subscribed to <i>BluetoothBeaconSumCnf</i>).
stateUnpluggedNum	integer	Number of unplugged Bluetooth beacons (when subscribed to <i>BluetoothBeaconSumState</i>).
statePluggedNum	integer	Number of plugged Bluetooth beacons (when subscribed to <i>BluetoothBeaconSumState</i>).
stateFailedNum	integer	Number of failed Bluetooth beacons (when subscribed to <i>BluetoothBeaconSumState</i>).

Available filters: (none)

4.62.17 BTStatisticDataType

This type contains all state information about a Bluetooth client. It is used in different requests and responses defined in this document.

Name	Type	Description
id	integer	Unique Bluetooth beacon identifier. The numbering starts at 0.

rssi	integer	Detected RSSI value to this Bluetooth client device (in dBm, e. g. – 75).
------	---------	---

4.62.18 BTClientStatisticDataType

This type contains all state information about a Bluetooth client. It is used in different requests and responses defined in this document.

Name	Type	Description
ppn	integer	Optional PPN id, if this Bluetooth client is a Bluetooth client in this DECT phone.
ethAddr	string	Ethernet address, format "00:11:22:aa:bb:cc".
btLocStat	sequence of BTStatisticDataType	List of all Bluetooth client locating statistic data.

4.62.19 GetBluetoothClientStatistic

With this request the client can acquire one or more information about the current Bluetooth locating statistics. The request contains an *ethAddr* attribute from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

This *ethAddr* usually start at "00:00:00:00:00:00",.

If the client asked for a certain number of records and a given *ethAddr*, the subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = "0", default), the given *ethAddr* has to match to a record exactly. If there is a record with this *ethAddr*, this one is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetBluetoothClientStatistic* (*ethAddr* = "00:00:00:00:00:00", *maxRecords* = "20"). If the response contains 20 records, take the last *ethAddr* send another *GetBluetoothClientStatistic*. Repeat this until you get less than 20 records back.

Following fields are defined for the request *GetBluetoothClientStatistic* in addition to the common attributes:

Name	Type	Mandatory	Description
<i>ethAddr</i>	string	yes	Bluetooth MAC address identifier to get the Bluetooth locating statistics got from. The numbering starts at "00:00:00:00:00:00".
<i>maxRecords</i>	integer, default = 1	no	Maximal number of records to return. Not more than 20 allowed..

The reply to *GetBluetoothClientStatistic* is an object called *GetBluetoothClientStatisticResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
<i>btClientStat</i>	sequence of BTClientStatisticDataType	yes	List of Bluetooth client statistic data.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Locating*

4.62.20 EventBluetoothClientStatistic

This event is sent by OM AXI when a buetooth client indicates new locating statistics. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *LocatingBluetooth*. This is allowed if the client has at least one of the following permissions: *Locating*.

The notification contains all current attributes of Bluetooth client locating statistics.

Attributes sent in this notification:

Name	Type	Mandatory	Description
btClientStat	BTClientStatisticDataType	yes	Current statistic data of the Bluetooth client that changes.

Available filters: no

4.63 Video Device Configuration And Maintenance

4.63.1 PpType

This type contains DECT phone element.

Name	Type	Description
ppn	integer	Portable part number.

4.63.2 VideoDevType

This type contains all configuration data fields of video device settings. It is used in different requests and responses defined in this document. Not all fields are used in all OMM versions and in all request and response types.

Name	Type	Description
id	integer	Unique video device identifier. The numbering starts at 0. This element is read only and cannot be set.
name	string	Unique video device name identifier.
usbPath	string	USB device path where the video device is detected. This element is read only and cannot be set.
tag	string	Internal name tag of the video device. This tag is sent by the video device when it is plugged in. The value can only be set in the <i>CreateVideoDev</i> request otherwise it is read only. Make sure, that this tag has to fit to the plugged video device name tag when created with a <i>CreateVideoDev</i> request, otherwise it would not be linked to the plugged video device. While the video device is automatically created, when it is plugged in, the administrator can configure the video data set afterwards with information of <i>EventVideoDevCnf</i> or <i>GetVideoDev</i> by using the <i>SetVideoDev</i> request.

rfpld	integer	RFP identifier the video device is connected to.
active	boolean	„1” or “true”, if this video device is enabled .
state	string	One of “unplugged”, “inactive”, “active” or “failed”. This element is read only and cannot be set.
resolution	string	One of “QCIF” (176*144 pixel), “QVGA” (320*240 pixel), “CIF” (352*288 pixel), “VGA” (640*480 pixel) or “SVGA” (800*600 pixel). Default setting is “SVGA”.
frameRate	integer	Frame rate in frames per seconds. Valid values are 5-10 frames per second. Default setting is 10.
hierarchy1	string	Position hierarchy level 1, free format text
hierarchy2	string	Position hierarchy level 2, free format text
hierarchy3	string	Position hierarchy level 3, free format text
hierarchy4	string	Position hierarchy level 4, free format text
ppnNum	integer	Number of DECT phones connected to this video device . This element is read only.
ppList	Sequence of PpType	List of all DECT phones connected to this video device. This element is read only.
siteId	integer	Site id this video device is connected to. This element is read only.
siteName	Sequence of PpType	Name of the site this video device is connected to. This element is read only.

4.63.3 GetVideoDev

With this request the client can query information about one or more video devices from the OMM.

The request contains an *id* from which starting subsequent data sets are to be fetched. The client can chose the number of records returned. If less than this number of data sets are returned or *ENoEnt*, the client knows that it reached end of file.

id s usually start at 0, but possibly not all numbers are assigned, i.e. the data set numbering may not be contiguous.

The responses are ordered by *id*, starting with the smallest.

If the client asked for a certain number records but the given *id* does not exist, subsequent entries will be returned.

If the client does not specify the number of records to be returned (*maxRecords* = “0”, default), the given *id* has to match to a record exactly. If there is a record with this *id*, it is returned, otherwise *ENoEnt*.

To acquire all data sets, one could send *GetVideoDev* (*id* = “0”, *maxRecords* = “20”). If the response contains 20 records, take the last *id* plus one and send another *GetVideoDev*. Repeat this until you get less than 20 records back.

Following fields are defined for the request *GetVideoDev* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	First video device id of video devices to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed. For special case 0 refer to text.

The reply to *GetVideoDev* is an object called *GetVideoDevResp*. It contains following element in addition to the common attributes:

Name	Type	Mandatory	Description
videoDev	sequence of VideoDevType	no	Video device record(s), if found

4.63.4 CreateVideoDev

The client can send this request to OM AXI to create a new video device data set. Note that attributes describing transient properties are ignored, e. g. *state*.

Following fields are defined for the request *CreateVideoDev* in addition to the common attributes:

Name	Type	Mandatory	Description
videoDev	VideoDevType	yes	Data of video device to create.

If the video device creation was successful, the reply contains all video device attributes which *GetVideoDev* would return. If the creation fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *CreateVideoDevResp*. It contains all actual set attributes, all attributes are optional.

For possible values for *errorCode*: see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.63.5 DeleteVideoDev

With this request the client can delete a video device record in the OMM. The key used to delete a video device is its *id*.

Consider the license restrictions for this command (see 5.5).

Following fields are defined for the request *DeleteVideoDev* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	id of video device to delete

The reply to *DeleteVideoDev* is an object called *DeleteVideoDevResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

For possible values for *errorCode*: see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.63.6 SetVideoDev

The client can send this request to OM AXI to change a video device data set. The *id* has to be filled in by the client to identify the record to be changed. Additionally the attributes of the video device which have to be changed must be filled in by the client.

Following fields are defined for the request *SetVideoDev* in addition to the common attributes:

Name	Type	Mandatory	Description
videoDev	VideoDevType	yes	Data of video device to change.

If the video device change was successful, the reply contains all video device attributes which *GetVideoDev* would return. If the creation fails, the OM AXI returns an error code. If applicable, the attributes will contain the data fields which lead to the error (which was wrong, missing etc.).

The reply to this request is called *SetVideoDevResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.63.7 EventVideoDevCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *VideoDev*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

This event is also sent upon creation of a video device. If the video device has been deleted, this event is sent with only an *id* in the *videoDev* element and the attribute *deleted="1"*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
videoDev	VideoDevType	yes	Contains all attributes which have been changed.
deleted	boolean, default false		„1” or “true”, if this data set has been deleted

4.63.8 GetVideoDevSummary

With this request the client can query common information about the video devices configured.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to this request is an object called *GetVideoDevSummaryResp*. It contains following fields in addition to the common attributes:

Name	Type	Description
nRecords	integer	The total number of video devices set up.
idFirst	integer	The ID of the first video device set up.
enabledNum	integer	Number of enabled video devices.
stateActiveNum	integer	Number of active video devices.
stateInactiveNum	integer	Number of inactive video devices.
stateUnpluggedNum	integer	Number of unplugged video devices.
stateFailedNum	integer	Number of failed video devices.

For possible `errCode` values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.63.9 EventVideoDevSummary

This event is sent by OM AXI when the number of video device summary counters have been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *VideoDevSumCnf* or to *VideoDevSumState*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Description
nRecords	integer	The total number of video devices set up (when subscribed to <i>VideoDevSumCnf</i>).
idFirst	integer	The ID of the first video device set up (when subscribed to <i>VideoDevSumCnf</i>).
enabledNum	integer	Number of enabled video devices.
stateActiveNum	integer	Number of active video devices.
stateInactiveNum	integer	Number of inactive video devices.
stateUnpluggedNum	integer	Number of unplugged video devices.
stateFailedNum	integer	Number of failed video devices (when subscribed to <i>VideoDevSumState</i>).

Available filters: (none)

4.63.10 GetVideoDevLink

With this request the client can query for the link of a video device id from the OMM.

Following fields are defined for the request *GetVideoDevLink* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	Video device identification to get the link to it from.
videoStream	boolean	yes	„1” or “true”, if a stream video link is requested. „0” or “false”, if a snapshot video link is requested.

The reply to *GetVideoDevLink* is an object called *GetVideoDevLinkResp*. It contains following element in addition to the common attributes:

Name	Type	Description
------	------	-------------

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *Video*

4.63.11 EventVideoDevLink

This event is sent by OM AXI when a video device link is ready. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *VideoDev*. This is allowed if the client has at least one of the following permissions: *Video*.

The notification contains all attributes which have been changed. Even attributes with default values will always be contained explicitly if their value has changed.

Attributes sent in this notification:

Name	Type	Description
id	integer	Video device identification of the delivered link.
videoStream	boolean	„1” or “true”, if a stream video link is delivered. „0” or “false”, if a snapshot video link is delivered.
videoDevLink	string	Link to the video device

4.64 Additional Settings Configuration And Maintenance

4.64.1 GetAdditionalSettings

With this request the client can query the configuration of the additional settings.

Following fields are defined for the request *GetAdditionalSettings* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply is an object called *GetAdditionalSettingsResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

truncatePpUserName	boolean	yes	„1” or “true”, if the truncating of the PPUser <i>name</i> attribute is allowed when the client sets or creates a too long name for the user to avoid an error for the request.
--------------------	---------	-----	---

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.64.2 SetAdditionalSettings

With this request the client can set the configuration of the additional settings.

Following fields are defined for the request *SetAdditionalSettings* in addition to the common attributes:

Name	Type	Mandatory	Description
truncatePpUserName	boolean	yes	„1” or “true”, to enable the truncating of the PPUser <i>name</i> attribute to the maximum length, if the client sets or creates a too long name for the user.

The reply is an object called *SetAdditionalSettingsResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.64.3 EventAdditionalSettingsCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
truncatePpUserName	boolean	no	„1” or “true”, if the truncating of the PPUser <i>name</i> attribute is allowed when the client sets or creates a too long name for the user to avoid an error for the request.

Available filters: (none)

4.65 Synchronization Monitoring

The DECT synchronization between the RFPs can be monitored using the requests, responses and events described in this chapter.

4.65.1 SyncOffsetType

This type contains the synchronization offset from an RFP to its neighbors. It has following attributes:

Name	Type	Description
id	integer	RFP ID of neighbor RFP
offset	integer	Time offset in ns
rsi	integer	RSSI in dBm, e. g. -75
lost	boolean, default = false	„1” or “true”, if this relation has been lost

If the attribute *lost* is set to *true*, this relation must be removed from the set of relations which may be stored at client side.

4.65.2 GetRFPSync

With this request a client can get the current synchronization state of an RFP.

Following fields are defined for the request *GetRFPSync* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	RFP id

The reply to *GetRFPSync* is an object called *GetRFPSyncResp*. It contains following attributes in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	RFP id
syncState	RFPSyncStateType	yes	State info
forward	sequence of SyncOffsetType	no	Relations to neighbor RFPs, if available
backward	sequence of SyncOffsetType	no	Relations from neighbor RFPs, if available

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.65.3 EventRFPSyncRel

This event is sent by OM AXI each time when the set of synchronization relations of an RFP changes. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *RFPSync*. This is allowed if the client has at least one of the following permissions: *Monitoring*.

Note that this event does only report changed forward relations. It is sent if there is a subscription for at least one of the two RFPs involved into the relation.

If a forward relation of one RFP changes, the client must imply a change of a backward relation of the other RFP. When an RFP state changes from *Synced* to any other state, the client must imply that all relations got lost in both directions. For this reason the client should also evaluate *EventRFPState* (see 4.42.8).

This event contains following fields:

Name	Type	Mandatory	Description
id	integer	yes	RFP-ID
forward	sequence of SyncOffsetType	no	Relations to neighbor RFPs, if available

Available filters: *rfp*

4.66 RFP Quality Monitoring

4.66.1 Media Stream Quality Monitoring

The RFP media stream quality can be monitored using the requests, responses and events described in this chapter.

4.66.1.1 MsQualityType

This type contains the media stream quality data of an RFP. It has following attributes:

Name	Type	Description
id	integer	RFP id
connects	integer	Number of audio connections this RFP has served so far. Note: For each handover there is a new audio connection.
duration	integer	Duration sum in seconds of all audio connections this RFP has served so far.
packetsRx	integer	Sum of RTP packets, this RFP has recieved while serving audio connections.
octetsRx	integer	Sum of audio data (not including header), this RFP has recieved while serving audio connections.
packetsTx	integer	Sum of RTP packets, this RFP has sent while serving audio connections.
octetsTx	integer	Sum of audio data (not including header), this RFP has sent while serving audio connections.
packetsLost	integer	Percent of packets considered as lost.
maxJitter	integer	Jitter measured in ms.

4.66.1.2 GetRFPMediaStreamQuality

With this request a client can get the current media stream quality data of an RFP.

Following fields are defined for the request *GetRFPMediaStreamQuality* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	First RFP id of RFP media stream quality data records to get

maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed.
------------	----------------------	----	--

The reply to *GetRFPMediaStreamQuality* is an object called *GetRFPMediaStreamQualityResp*. It contains following attributes in addition to the common attributes:

Name	Type	Mandatory	Description
msQuality	sequence of MsQualityType	yes	All found media stream quality data records

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead, Monitoring*

4.66.1.3 ResetRFPMediaStreamQuality

With this request a client can reset the current media stream quality data of one or all RFPs.

Following fields are defined for the request *ResetRFPMediaStreamQuality* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	RFP id of RFP media stream quality data records to reset. If -1 is given, the media stream quality data records of all RFPs will be reset.

The reply to *ResetRFPMediaStreamQuality* is an object called *ResetRFPMediaStreamQualityResp*. It contains following attributes in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite, Monitoring*

4.66.1.4 EventRFPMediaStreamQuality

This event is sent by OM AXI each time when the set of media stream quality data of an RFP changes. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *RFPsQuality*. This is allowed if the client has at least one of the following permissions: *Monitoring*.

This event contains following fields:

Name	Type	Mandatory	Description
msQuality	sequence of MsQualityType	yes	Media stream quality data records

Available filters: *rfp*

4.66.2 IP Quality Monitoring

The RFP IP quality can be monitored using the requests, responses and events described in this chapter.

4.66.2.1 IpQualityType

This type contains the IP quality data of an RFP. It has following attributes:

Name	Type	Description
id	integer	RFP id
connectedTime	integer	Time the RFP is connected to the OMM in seconds.
currentRTT	integer	Current round trip time (rtt) in μ s.
maxRTT	integer	Highest rtt measured for this RFP in μ s.
count	integer	Number of rtt samples acquired so far.
interval1	integer	Number of rtt samples within interval 1 limits
interval2	integer	Number of rtt samples within interval 2 limits
interval3	integer	Number of rtt samples within interval 3 limits
interval4	integer	Number of rtt samples within interval 4 limits
interval5	integer	Number of rtt samples within interval 5 limits

Note: For interval limits see *Limits* in .4.1.3.

4.66.2.2 GetRFPIpQuality

With this request a client can get the current IP quality data list of RFPs.

Following fields are defined for the request *GetRFPIpQuality* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	First RFP id of RFP IP quality data records to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed.

The reply to *GetRFPIpQuality* is an object called *GetRFPIpQualityResp*. It contains following attributes in addition to the common attributes:

Name	Type	Mandatory	Description
ipQuality	sequence of IpQualityType	no	All found IP quality data records

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*, *Monitoring*

4.66.2.3 EventRFPIpQuality

This event is sent by OM AXI each time when the set of IP quality data of at least one RFP changes. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *RFPIpQuality*. This is allowed if the client has at least one of the following permissions: *Monitoring*.

This event contains following fields:

Name	Type	Mandatory	Description
ipQuality	sequence of IpQualityType	yes	IP quality data records

4.66.3 Synchronization Quality Monitoring

The RFP synchronization (SYNC) quality can be monitored using the requests, responses and events described in this chapter.

4.66.3.1 SyncQualityType

This type contains the SYNC quality data of an RFP. It has following attributes:

Name	Type	Description
id	integer	RFP id
strongRels	integer	Number of relations with an RSSI-value greater than -73dBm
lowRels	integer	Number of relations with an RSSI-value less than -73dBm
maxRSSI	integer	Maximum RSSI value (in dBm) of the strong relations
minRSSI	integer	Minimum RSSI value (in dBm) of the low relations

4.66.3.2 GetRFPSyncQuality

With this request a client can get the current SYNC quality data list of RFPs.

Following fields are defined for the request *GetRFPSyncQuality* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	First RFP id of RFP SYNC quality data records to get
maxRecords	integer, default = 0	no	Maximal number of records to return. Not more than 20 allowed.

The reply to *GetRFPSyncQuality* is an object called *GetRFPSyncQualityResp*. It contains following attributes in addition to the common attributes:

Name	Type	Mandatory	Description
syncQuality	sequence of SyncQualityType	no	All found SYNC quality data records

For possible `errCode` values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead, Monitoring*

4.66.3.3 EventRFPSyncQuality

This event is sent by OM AXI each time when SYNC quality data of at least one RFP changes. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *RFPSyncQuality*. This is allowed if the client has at least one of the following permissions: *Monitoring*.

This event contains following fields:

Name	Type	Mandatory	Description
syncQuality	sequence of SyncQualityType	yes	SYNC quality data records

4.67 DECT Phone Device Profile Configuration And Maintenance

The DECT phone profile configuration data will be used and loaded into the DECT phone when the following conditions can be applied for a DECT phone device data set linked to a DECT phone user data set:

- DECT phone firmware supports a profile configuration (certain HW is required) / see also DECT phone device capability flag for supporting this feature.
- The DECT phone is connected to its “master” (OMM) system
-
- The DECT phone firmware has the correct version, same as OMM or newer
- The DECT phone has identified its current loaded profile configuration setting during the location registration
- The DECT phone has no or a different profile configuration loaded than required
- The user data set is enabled to load the DECT phone configuration data and has a configuration profile id set to be loaded.

A merge with the DECT phone default device profile configuration (`id="0"` `name="Default"`) data and user specific DECT phone configuration data might be performed

The DECT phone profile configuration data will be merged and loaded into the DECT phone when the user logs in at the DECT phone. The default configuration data set (`id="0"` `name="Default"`) will be loaded into the DECT phone, when the user logged out from the DECT phone or the device does a location registration after subscription to an unbound state to a user.

Note: The system performs a sequential loading of configuration data to the DECT phones. Therefore updating of all DECT phones in a system with a high number of DECT phones might be last a long time depending on the traffic on the DECT air interface.

4.67.1 PpProfileDataSetType

This type contains all DECT phone device profile configuration data fields. It is used in different requests and responses defined in this document.

Name	Type	Mandatory	Description
name	string	yes	Name of the DECT phone profile configuration data. Note: The value <i>name</i> ="Default" is a reserved name for the default device DECT phone profile configuration data and cannot be set.
id	integer	no	ID of the DECT phone profile configuration data set. This value is in a range from 1 to <i>ppProfileNum</i> (20) to be set. Up to <i>ppProfileNum</i> (20) user group DECT phone configuration profiles can be used for users (see 4.39.1). The value <i>id</i> ="0" is a reserved id for the default device DECT phone profile configuration data (see 4.38.1).
ppData	string	yes	Configuration data to be downloaded to the DECT phone. The data consists of a sequence of lines containing one setting per line. E. g.: # display-einstellungen UD_Displanguage=en UD_DisplFont=large UD_DisplColor=black # ringer-einstellungen UD_RingerVolumeIntern=level-1 Maximal size is 4 kByte. If <i>ppData</i> ="" the dataset is deleted or should be deleted for a set operation.
timeStamp	long integer	no	Last change time stamp of this element in seconds since 1970/1/1. When this element is not used in a set request the element is only changed when the data record differs from the record already stored in the OMM database,

4.67.2 GetPpProfile

With this request the client can query for a DECT phone profile configuration data set.

Following fields are defined for the request *GetPpProfile* in addition to the common attributes:

Name	Type	Mandatory	Description
id	integer	yes	ID of the DECT phone device profile configuration data set.

The reply is an object called *GetPpProfileResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
ppProfile	PpProfileDataSetType	yes	Requested DECT phone profile configuration data sets.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.67.3 SetPpProfile

With this request the client can set the DECT phone device profile configuration data.

Following fields are defined for the request *SetPpProfile* in addition to the common attributes:

Name	Type	Mandatory	Description
ppProfile	PpProfileDataSetType	yes	The DECT phone profile configuration data to be set.

The reply is an object called *SetPpProfileResp*. It contains all actual set DECT phone device profile configuration data.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.67.4 EventPpProfileCnf

This event is sent by OM AXI when a DECT phone device profile configuration has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *PPHCM*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
deleted	boolean	no	„1” or “true”, DECT phone device profile configuration data has been deleted.
ppProfile	PpProfileDataSetType	no	The new or changed DECT phone device profile configuration data.

Available filters: (none)

4.68 Core Dump Configuration And Maintenance

4.68.1 GetCoreDumpURL

With this request the client can query the system configuration URL settings used to download core dump files. Following fields are defined for the request *GetCoreDumpURL* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to download core dump files to the external server.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.68.2 SetCoreDumpURL

With this request the client can set the system configuration URL settings used to download core dump files.

Following fields are defined for the request *SetCoreDumpURL* in addition to the common attributes:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to download core dump files to the external server.

The reply is an object called *SetCoreDumpURLResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.68.3 EventCoreDumpURLCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
url	URLType	yes	Configurable URL settings to download core dump files to the external server.

Available filters: (none)

4.69 NTP Server Configuration And Maintenance

4.69.1 GetNTPServer

With this request the client can query the system NTP server settings.

Following fields are defined for the request *GetNTPServer* in addition to the common attributes:

Name	Type	Mandatory	Description
ntpServerName1	string	no	First NTP server name to use.
ntpServerName2	string	no	Second NTP server name to use.
ntpServerName3	string	no	Third NTP server name to use.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.69.2 SetNTPServer

With this request the client can set the system NTP server settings.

Following fields are defined for the request *SetNTPServer* in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

default	boolean	no	„1” or “true”, if the NTP server settings are reset to NTP default settings. The other parameters of this request are ignored.
ntpServerName1	string	no	First NTP server name to use. Used when <i>default</i> ="0".
ntpServerName2	string	no	Second NTP server name to use. Used when <i>default</i> ="0".
ntpServerName3	string	no	Third NTP server name to use. Used when <i>default</i> ="0".

The reply is an object called *SetNTPServerResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.69.3 EventNTPServerCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
ntpServerName1	string	no	First NTP server name to use.
ntpServerName2	string	no	Second NTP server name to use.
ntpServerName3	string	no	Third NTP server name to use.

Available filters: (none)

4.70 Special Branding Configuration And Maintenance

4.70.1 GetSpecialBranding

With this request the client can query the configuration of Mitel SIP-DECT special branding.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Description
		(none)

The reply is an object called *GetSpecialBrandingResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
key	string	yes	Against this key branding of DECT phones is validated. If this value is set, it is 32 bytes long and only DECT phones can be

			subscribed with appropriate branding setting.
url	URLType	yes	Server configuration settings where the branding image file can be gotten.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.70.2 SetSpecialBranding

With this request the client can configure the Mitel SIP-DECT special branding.

Only attributes which have to be changed need to be specified.

Following fields are defined for the request *SetSpecialBranding* in addition to the common attributes:

Name	Type	Mandatory	Description
key	string	no	Against this key branding of DECT phones is validated. If this value is set, only DECT phones can be subscribed with appropriate branding setting. When set, this key is 64 bytes long, but since the second half is just a checksum against incorrect entries, only 32 bytes are stored in the database. There is also be a 32-byte key necessary to delete the DECT phone branding validation.
url	URLType	no	Server configuration settings where the branding image file can be gotten.

If the change was successful, the reply contains all attributes which *GetSpecialBranding* would return.

The reply is an object called *SetSpecialBrandingResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.70.3 EventSpecialBrandingCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *SystemCnf*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

Attributes sent in this notification:

Name	Type	Mandatory	Description
key	string	no	Against this key branding of DECT phones is validated. If this value is set, it is 32 bytes long and only DECT phones can be subscribed with appropriate branding setting.
url	URLType	no	Server configuration settings where the branding image file can be gotten.

Available filters: (none)

4.71 User Device Synchronization Configuration And Maintenance

This configuration settings are provided to specify a central OMM for several OMM sites within the SIP-DECT system doing the user device synchronization for roaming DECT phones between the OMM sites.

4.71.1 GetUserDeviceSyncOMM

This request is sent from the client to the OMM.

With this request the client can query common information about the central OMM configuration of user device synchronization for roaming .

Following fields are defined for the request *GetUserDeviceSyncOMM* in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetUserDeviceSyncOMM* is an object called *GetUserDeviceSyncOMMResp*. It contains following fields in addition to the common attributes:

Name	Type	Mandatory	Description
enable	boolean	yes	„1” or “true”, if an OMM is configured within a SIP DECT system providing the user device synchronization for DECT phone roaming between OMMs.
omm1DomainName	string	yes	First domain name or IP address of the OMM within a SIP DECT system providing the user device synchronization.
omm2DomainName	string	yes	Second domain name or IP address of the OMM within a SIP DECT system providing the user device synchronization. This is necessary when the OMM site supporting the user device synchronization runs in a redundant or standby OMM constallation.
user	string	yes	OMM admin user account to access the OMM providing the user device synchronization.
password	string	yes	OMM admin user account to access the OMM providing the user device synchronization, encrypted with public key.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.71.2 SetUserDeviceSyncOMM

This request is sent from the client to the OMM.

With this request the client can set the central OMM configuration of user device synchronization for roaming configuration settings.

Following fields are defined for the request *SetUserDeviceSyncOMM* in addition to the common attributes:

Name	Type	Mandatory	Description
------	------	-----------	-------------

enable	boolean	no	„1” or “true”, if an OMM is configured within a SIP DECT system providing the user device synchronization for DECT phone roaming between OMMs.
omm1DomainName	string	no	First domain name or IP address of the OMM within a SIP DECT system providing the user device synchronization.
omm2DomainName	string	no	Second domain name or IP address of the OMM within a SIP DECT system providing the user device synchronization. This is necessary when the OMM site supporting the user device synchronization runs in a redundant or standby OMM constallation.
user	string	no	OMM admin user account to access the OMM providing the user device synchronization.
password	string	no	OMM admin user account to access the OMM providing the user device synchronization, encrypted with public key.

The reply to *SetUserDeviceSyncOMM* is an object called *SetUserDeviceSyncOMMResp*. It contains all actual set attributes, all attributes are optional.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*

4.71.3 EventUserDeviceSyncOMMCnf

This event is sent by OM AXI when this configuration item has been changed. Refer to chapter 4.1.4 on how to set up a notification session.

To get this event the client has to subscribe to *Roaming*. This is allowed if the client has at least one of the following permissions: *AllCnfRead*.

The notification contains all attributes which have been changed.

Attributes sent in this notification:

Name	Type	Mandatory	Description
enable	boolean	no	„1” or “true”, if an OMM is configured within a SIP DECT system providing the user device synchronization for DECT phone roaming between OMMs.
omm1DomainName	string	no	First domain name or IP address of the OMM within a SIP DECT system providing the user device synchronization.
omm2DomainName	string	no	Second domain name or IP address of the OMM within a SIP DECT system providing the user device synchronization. This is necessary when the OMM site supporting the user device synchronization runs in a redundant or standby OMM constallation.

user	string	no	OMM admin user account to access the OMM providing the user device synchronization.
password	string	no	OMM admin user account to access the OMM providing the user device synchronization, encrypted with public key.

Available filters: (none)

4.72 System Statistic Counters Management

There are several system statistic counters implemented. Each counter will be identified by a unique element id. These ids (counters) are arranged in two different groups:

- Occurrence counter
- MinMax counter

The occurrence counter only counts the occurrence of events.

The MinMax counter counts the occurrence, the minimum and maximum values, the sum and optional the frequency distribution of measured values.

All counters may have a history of n records and the history time-unit can be read out of the OMM.

4.72.1 SysStatNameType

This type contains information about the single system statistic elements (counter).

Name	Type	Mandatory	Description
elemId	integer	yes	Unique element-id of a statistic counter
type	string	yes	Type of element: <ul style="list-style-type: none"> • “occurrence” • “minmax”
name	string	yes	Name and meaning of the current element in English
resolution	string	yes	Resolution of the history records of this element <ul style="list-style-type: none"> • “min” • “hour” • “day” • “week”
duration	integer	yes	Duration * resolution defines the time length for one history records
numOfRecords	integer	yes	Number of history records
sum	integer	no	Sum counter – Only “occurrence” counter
unit	string	no	Unit of the counted value – Only “minmax” counter
numOfFreqDistUnits	integer	no	Number of units for an optional frequency distribution curve – Only “minmax” counter

4.72.2 GetSysStatisticConfig

This request is sent from the client to the OMM.

With this request the client can query information about the system statistic counter elements. To get an overview of the existing system statistic counters this request can be used.

Following fields are defined for the request *GetSysStatisticConfig*:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetSysStatisticConfig* is an object called *GetSysStatisticConfigResp*. It contains the following element for each existing system statistic counter:

Name	Type	Mandatory	Description
sysStatName	SysStatNameType	yes	Details of a single element

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.72.3 GetSysStatisticOccurrence

This request is sent from the client to the OMM.

With this request the client can query information about the system statistic counter elements.

Following fields are defined for the request *GetSysStatisticOccurrence*:

Name	Type	Mandatory	Description
elemId	integer	yes	Unique element id
firstRecord	integer	no	First history record
maxRecords	integer	no	Max records in answer

The reply to *GetSysStatisticOccurrence* is an object called *GetSysStatisticOccurrenceResp*. It contains the following fields:

Name	Type	Mandatory	Description
elemId	integer	yes	Unique element id
age	integer	yes	Age of the sum field in duration * resolution
sum	integer	yes	Sum counter
firstRecord	integer	no	First history record if requested
counter	integer list	yes	List of all requested history records
currMeasuredTime	integer	no	Measured time in sec for the current record. This field only exists if the current record was requested
oldMeasuredTime	integer	no	Measured time in sec for the oldest record. This field only exists if the oldest record was not full measured

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*, *Monitoring*

4.72.4 GetSysStatisticMinMaxSummary

This request is sent from the client to the OMM.

With this request the client can query information about the system statistic counter elements.

Following fields are defined for the request *GetSysStatisticMinMaxSummary*:

Name	Type	Mandatory	Description
elemId	integer	yes	Unique element id

The reply to *GetSysStatisticMinMaxSummary* is an object called *GetSysStatisticMinMaxSummaryResp*. It contains the following fields:

Name	Type	Mandatory	Description
elemId	integer	yes	Unique element id
age	integer	yes	Age of the sum field in duration * resolution
min	integer	yes	Minimum counter value
max	integer	yes	Maximum counter value
sum	integer	yes	Sum counter
occurrence	integer	yes	Number of events counted
freqDistUnits	integer list	no	Frequency distribution curve units
freqDistValues	integer list	no	Frequency distribution values

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: AllCnfRead, Monitoring

4.72.5 SysRecHeaderType

This type contains information about a single minmax-element-record.

Name	Type	Mandatory	Description
elemId	integer	yes	Unique element id of a record-element
counterName	string	yes	Comma separated name list. These are the names of the counters in the SysRecType
firstRecord	integer	no	First history record number
currMeasuredTime	integer	no	Measured time in sec for the current record. This field only exists if the current record was requested
oldMeasuredTime	integer	no	Measured time in sec for the oldest record. This field only exists if the oldest record was not full measured

4.72.6 SysRecType

This type contains the counter of a single record of a minmax-element.

Name	Type	Description
rec	integer	record number
cnt	string	Comma separated value list. These are the values of the record counters

4.72.7 GetSysStatisticMinMaxRecord

This request is sent from the client to the OMM.

With this request the client can query information about the system statistic counter elements.

Following fields are defined for the request *GetSysStatisticMinMaxRecord*

Name	Type	Mandatory	Description
elemId	integer	yes	Unique element id
firstRecord	integer	no	First history record number
maxRecords	integer	no	Max records in answer

The reply to *GetSysStatisticMinMaxRecord* is an object called *GetSysStatisticMinMaxRecordResp*. It contains the following fields:

Name	Type	Mandatory	Description
sysRecHeader	SysRecHeaderType	yes	
sysRec	SysRecType	yes	

For possible `errCode` values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*, *Monitoring*

4.72.8 GetSysStatisticMinMax

This request is sent from the client to the OMM.

With this request the client can query history records for a single counter-value of a minmax-counter .

Following fields are defined for the request *GetSysStatisticMinMax*

Name	Type	Mandatory	Description
elemId	integer	yes	Unique element id
type	string	yes	Counter-type: <ul style="list-style-type: none"> • “min” • “max” • “sum” • “occurrence”
firstRecord	integer	no	First history record
maxRecords	integer	no	Max records in answer

The reply to *GetSysStatisticMinMax* is an object called *GetSysStatisticMinMaxResp*. It contains the following fields:

Name	Type	Mandatory	Description
elemId	integer	yes	Unique element id
type	string	yes	Requested type
firstRecord	integer	no	First history record
counter	string	yes	Comma separated value list. These are the values of the requested history record counter
currMeasuredTime	integer	no	Measured time in sec for the current record (record 0). This field only exists if the current record was requested
oldMeasuredTime	integer	no	Measured time in sec for the oldest record. This field only exists if the oldest record was not full measured

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*, *Monitoring*

4.72.9 ResetSysStatistic

This request is sent from the client to the OMM.

With this request the client reset the sum counter of a single element or of all elements.

Following fields are defined for the request *ResetSysStatistic*

Name	Type	Mandatory	Description
elemId	integer	yes	<ul style="list-style-type: none"> • Unique element id or • -1 means all system statistic elements

The reply to *ResetSysStatistic* is an object called *ResetSysStatisticResp*. It contains no fields.

The client needs one of these permissions to use this request: AllCnfWrite, Monitoring

4.73 RFP Statistic Counters Management

The following statistic counters are implemented for each RFP. These counters are defined with a unique RFP statistic element id:

Element Id	Description
Voice channels utilization	
0	Only 2 voice channels available
1	Only 1 voice channel available
2	No more voice channels available
3	Setup rejected because of not available voice channel
Air Interface utilisation	
4	After a new connection there are only 2 or 3 free air channels available
5	Only 1 air channel available
6	No more air channels available
Paging	
7	Paging queue overflow. A prior page request may be overwritten by a new page request
Synchronization	
8	The RFP has lost the synchronization to the system
9	The number of sync-relations to other RFPs has changed from > 2 to 1
10	Sync-Adjustment with ± 1 Bit
Health	
11	RFP reset
12	RFP not reachable – (no reset)

The counters can be read and reset by the OM AXI-Messages described in the subsequent chapters.

4.73.1 RFPStatHeadType

This type contains information about the RFP statistic elements (counter). The different elements are combined to a record set. Record set 0 always exists, it contains the overall counter. History record sets may be exists.

Name	Type	Description
numElemPerRec	integer	Number of different elements per RFP record
recordSets	integer	Number of history record sets Record 0 identifies the overall counter, 1 the current week, 2 the week before the current week and so on
resolution	string	resolution of the history data sets

4.73.2 RFPStatNameType

This type contains the information about a single RFP statistic element.

Name	Type	Description
elemId	integer	Unique element id of a record-element
group	string	String to identify the different element groups
name	string	Name and meaning of the current element in English

4.73.3 GetRFPStatisticConfig

This request is sent from the client to the OMM.

With this request the client can query information about the RFP statistic elements.

Following fields are defined for the request *GetRFPStatisticConfig*:

Name	Type	Mandatory	Description
			(no additional fields defined)

The reply to *GetRFPStatisticConfig* is an object called *GetRFPStatisticConfigResp*. It contains the following elements:

Name	Type	Mandatory	Description
rfpStatHead	RFPStatHeadType	yes	Summary header of the RFP statistic records
rfpStatName	RFPStatNameType	yes	Details of record-elements

For possible errCode values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.73.4 GetRFPStatistic

This request is sent from the client to the OMM.

With this request the client can query the statistic counter of an RFP.

Following fields are defined for the request *GetRFPStatisticConfig*:

Name	Type	Description
id	integer	Unique RFP identifier. The numbering starts at 0
maxRecords	integer	Maximal number of records to return. Not more than 20 allowed. If maxRecord is equal 0, only the record of the RFP addressed by id should be fetched
recordSet	integer	Record set to read Record 0 identifies the overall counter, 1 the current week, 2 the week before the current week and so on.

The type *RFPStatDataType* contains the elements of one record set of a single RFP.

Name	Type	Description
id	integer	Unique RFP identifier. The numbering starts at 0
counter	string	This string contains all elements of one RFP statistic record as a comma separated list.

The reply to *GetRFPStatistic* is an object called *GetRFPStatisticResp*. It contains the following elements:

Name	Type	Mandatory	Description
rfpStatData	RFPStatDataType	yes	Summary of the RFP statistic elements. The number and meaning of the elements can be read by using the <i>GetRFPStatisticConfig</i> Request.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*, *Monitoring*

4.73.5 ResetRFPStatistic

This request is sent from the client to the OMM.

With this request the client can reset the statistic overall counter(s) of one or more RFPs.

Following fields are defined for the request *ResetRFPStatistic*:

Name	Type	Description
id	integer	Unique RFP identifier. The numbering starts at 0 -1 means all RFPs
elemId	integer	Unique element id -1 means all elements of a record

For possible `errCode` values see 3.3.

The client needs one of these permissions to use this request: *AllCnfWrite*, *Monitoring*

The reply to *ResetRFPStatistic* is an object called *ResetRFPStatisticResp*. It contains the no fields.

4.74 File Transfer Management

4.74.1 GetFile

With this request a client can download a file from the OMM.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
name	string	yes	File name

The reply to this request is an object called *GetFileResp*. It contains following attributes in addition to the common attributes:

Name	Type	Mandatory	Description
name	string	no	File name, for reference reasons
offset	integer	no	File offset, for reference reasons
data	string	no	Base64 encoded file data block
size	integer	no	Overall size of file, if known. Only sent in the first response.
eof	boolean	no	„1” or “true”, if this is the last part of the file

By sending *GetFile* with a new file name, a client opens a file for downloading. It will remain open until it is read until EOF, until the client sends *GetFile* with a different file name or until the client gets disconnected.

A file is downloaded block wise. The size of the blocks is not specified, it may fill the XML messages up to the maximal size of OM AXI messages.

An example file transfer may look like this:

- The client asks *GetFile*("/foo/bar")
- OM AXI responds *GetFileResp*("/foo/bar", 0, <64 encoded bytes>)
- The client asks for the next block using *GetFile*("/foo/bar")
- OM AXI responds *GetFileResp*("/foo/bar", 64, <8 encoded bytes>, EOF)

- Client knows that it has received the complete file, because it got EOF.

A client can only download one file at a time. As soon as different file name is requested, the old file is closed, if any, and the new file is opened for reading.

Note that the *size*, which may be sent in the first reply when it is known, is for information only and does not replace *eof*. The client should be aware that there may be situations where the actual number of bytes may differ from the size which was told initially.

The file name can be a physical file on the OMM file system. Additionally there are logical names starting with a colon. These are the supported file names:

Value	Description
/...	Any file name starting with a slash points to the local file system in the OMM, e. g. "/foo/bar"
:config	Load the configuration file
:empty	Load an empty dummy file, will return 0 bytes and EOF, this can be used to force a previously opened file to be closed
:eula	Load the end user license agreement
:sysdump	Load a complete system dump containing various service information
:park	Load PARK XML file required to get a PARK configuration file from the park server.

The *data* field may be empty (""), e. g. for sending *eof* when there is no more data to be sent.

Logical files may cause the special error *EWouldBlock*. This means that there is no data available for this file currently, but the file did not reach EOF yet. The client can retry some seconds later.

Some files may be currently locked by another software or client. This may cause the error *EFailed*. In this case the client can try again to get the file later.

After getting *EFailed* or *EWouldBlock* the client should wait at least 5 seconds before trying again.

Possible values for *errCode*:

Value	Description
ENoEnt	File not found
EFailed	File is currently locked by another software or client, try again some seconds later.
EWouldBlock	There is no data available currently but the file did not reach EOF. This may only happen with logical files.
EPerm	This file cannot be downloaded with the currently active account.
ENoMem	An unspecified error happened while reading the file.

For further possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: *AllCnfRead*

4.74.2 PutFile

With this request a client can upload a file to the OMM.

Following fields are defined for this request in addition to the common attributes:

Name	Type	Mandatory	Description
name	string	yes	File name
offset	integer	yes	File offset, for reference reasons
data	string	yes	Base64 encoded file data block Note: In some cases data might be empty (e. g. for <i>:ima</i>), then the file will be deleted.
eof	boolean	no	„1” or “true”, if this is the last part of the file

The reply to this request is an object called *PutFileResp*. It contains following attributes in addition to the common attributes:

Name	Type	Mandatory	Description
			(no additional fields defined)

By sending *PutFile* with a new file name and *offset="0"*, a client opens a file for uploading. It will remain open until the client sends EOF or until an error occurs.

A file is uploaded block wise. The size of the blocks is not specified, it may fill the XML messages up to the maximal size of OM AXI messages, and however, a message size less than 1500 bytes is recommended to allow fast responses.

If the client starts to upload another file before the old one is completed, if the connection gets dropped or an error occurs, the partly uploaded file is silently discarded by the OMM.

The file name can be a physical file on the OMM file system. Additionally there are logical names starting with a colon. These are the supported file names:

Value	Description
/...	Any file name starting with a slash pointers to the local file system in the OMM, e. g. <i>"/foo/bar"</i> Note: This option is currently not supported by the OMM
:config	Upload and activate an OMM configuration file
:license	Upload and activate an OMM license file
:ima	Upload and activate an IMA configuration file
:park	Upload PARK configuration file for setting up a new PARK

The *data* field may be empty (""), e. g. for sending *eof* when there is no more data to be sent.

The client should be aware that each uploaded block may cause an error in the response. Once an error occurred, the file transfer must be considered as failed and being closed.

To cancel a running file transfer and to discard the data already uploaded, the client can send a possibly empty block of data with an invalid offset, e. g. *"-1"*. Some files may be currently locked by another software or client. This may cause the error *EFailed*. In this case the client can try again to get the file later.

When uploading a configuration file, the last response will return a positive response and restart the OMM if the configuration file is valid. If the configuration file cannot be used, *EInval* is returned for the attribute *data*.

When uploading a license file, the last response will return a positive response and activate the contained license, if the configuration file is valid. Depending on the license the OMM might restart. If the license file cannot be used, *EInval* is returned for the attribute *data*.

After getting EFailed the client should wait at least 5 seconds before trying again.

Possible values for *errCode*:

Value	Description
EInval	A field contains invalid data or exceeds a limit; this may also mean that the offset is out of sync. For License file the PARK is invalid.
EFailed	File is currently locked by another software or client, try again some seconds later.
ETooLong	A string is too long. If this is sent for the attribute <i>offset</i> , the file is too large.
EPerm	This file cannot be uploaded with the currently active account.
ENoMem	An unspecified error happened while writing the file. For License file the system is completely configured.
ELicenseFile	The license file to be put/imported is invalid
EForbidden	License file import is not allowed.
ELicenseInstallId	The configured installation ID does not fit to the license file.

For possible *errCode* values see 3.3.

The client needs one of these permissions to use this request: AllCnfWrite

5 Appendix – OpenMobility® Type Differences

5.1 Requests/Events Dependencies Of “OpenResp“-Parameters

The content or functionality of the following requests/events depends on the OMM capabilities sent in the “OpenResp”:

REQUESTS / EVENTS	Dependency to “OpenResp” attributes
All configuration requests	“haveConfigPython” For more information see detailed list in 5.2.
“AddUserToConference” “EventAddUserToConference”	“haveConference” “haveSRTP”
“DBBackupToUSB” “DBRestoreFromUSB”	“haveConfigPython”
“ConferenceConfirmation”	“haveConference”
“Create/Delete/Get/Set/PpProfile” “EventPpProfileCnf”	”haveHCM”
“Create/Delete/PPDev”	”haveUnboundDevices” “haveBluetooth” “haveUserDeviceSync”
“Create/Delete/Get/SetXMLApplication” “EventXMLApplicationCnf”	“haveXML” “haveXMLDynamic” “haveXMLCorpDir” “haveProtFTP” “haveProtFTPS” “haveProtHTTP” “haveProtHTTPS” “haveProtSCP” “haveProtSFTP” “haveProtTFTP”
“Create/DeleteACLEntry”	“haveWLAN”
“Create/DeleteConference”	“haveConference”
“Create/DeleteDigitTreatment”	“haveDigitTreatment”
“Create/DeleteLDAP”	“haveLDAP”
“Create/DeletePPUser” “CreateFixedPP”	“haveExternalUserData” ”haveSIP” ”haveSOSNumberPP” ”haveUnboundDevices” “haveVoiceboxNumber” “haveConference” “haveBluetooth” “haveVideo”

	“haveHCM”
“Create/DeleteWLANProfile”	“haveWLAN”
“Create/Delete/Get/SetSite” “EventSiteCnf”	“haveSRTP” “haveVideo”
“GetSiteSummary” “EventSiteSummary”	“haveSRTP”
“DeleteUserFromConference” “EventDeleteUserFromConference”	“haveConference”
“EventUserDataImport”	“haveExternalUserData” “haveUnboundDevices”
“EventACLCnf”	“haveWLAN”
“EventConferenceRelease”	“haveConference”
“EventConferenceRequest”	“haveConference”
“EventCreateConference”	“haveConference”
“EventDeleteConference”	“haveConference”
“EventWLANClient”	“haveWLAN”
“Get/EventReadyForConferencing”	“haveConference”
“Get/Set/Create/DeleteBluetoothBeacon” “EventBluetoothBeaconCnf” “GetBluetoothBeaconSummary” “EventBluetoothBeaconSummary” “Get/EventBluetoothClientStatistic” “Get/SetBluetoothGlobalSettings” “EventBluetoothGlobalSettingsCnf” “Get/SetBluetoothSensitivity” “EventBluetoothSensitivityCnf”	“haveBluetooth”
“Get/SetACLEntry”	“haveWLAN”
“Get/SetAdditionalSettings” “EventAdditionalSettingsCnf”	“haveAdditionalSettings”
“Get/SetAdvancedSIP” “EventAdvancedSIPCnf”	“haveSIP”
“Get/SetAutoDBBackup”, “EventAutoDBBackupCnf”	“haveAutoDB” “haveProtFTP” “haveProtFTPS” “haveProtHTTP” “haveProtHTTPS” “haveProtSCP” “haveProtSFTP”

	“haveProtTFTP”
“Get/SetBackupSIP” “EventBackupSIP”	“haveSIP”
“Get/SetBasicSIP” “EventBasicSIPCnf”	“haveSIP”
“Get/SetSecureSIP” “EventSecureSIPCnf” “GetSecureSIPCertificate” “SetSecureSIPCertificate” “EventSecureSIPCertificateCnf” “GetSecureSIPCertificateServerImport” “SetSecureSIPCertificateServerImport” “EventSecureSIPCertificateServerImportCnf”	“haveSIP”
“Get/SetPortRangeSIP” “EventPortRangeSIPCnf”	“haveSIP”
“Get/EventHealthState”	“haveHCM”
“Get/SetConferenceRoom” “EventConferenceRoomCnf”	“haveConference”
“Get/SetConferenceServerSIP” “EventConferenceServerSIPCnf”	“haveConference”
“Get/SetDECTPpSettings” “EventDECTPpSettingsCnf”	“haveDECTPpSettings”
“Get/SetDECTPagingAreaSize” “EventDECTPagingAreaSizeCnf”	“havePagingAreas”
“Get/SetDECTRegDomain” “EventDECTRegDomainCnf”	“haveDECTRegDomain”
“Get/SetDevAutoCreate” “EventDevAutoCreateCnf”	”haveUnboundDevices”
“Get/SetDigitTreatment” “EventDigitTreatmentCnf”	“haveDigitTreatment” “haveDigitTreatmentSimple”
“Get/SetDTMF” “EventDTMF”	“haveSIP”
“Get/SetFAC” “SetFACList” “EventFACCnf”	“haveFACs” ”haveUnboundDevices”
“Get/SetFACPrefix” “EventFACPrefixCnf”	“haveFACs”
“Get/SetIMA” “EventIMACnf”PutFile	“haveIMA”

“Get/SetIntercomCallHandlingSIP” “EventIntercomCallHandlingSIPCnf”	“haveSIP”
“Get/SetLDAP” “EventLDAPCnf”	“haveLDAP”
“GetLicense” ”EventLicenseCnf” “Get/SetLicenseServerList” “EventLicenseServerListCnf” “EventLicenseFile”	“haveLicense”
“Get/SetLoginVariant” “EventLoginVariantCnf”	“haveUnboundDevices” and “haveSIP”
“Get/SetNetParams” “EventNetParams”	“haveIPParamCnf” “haveWLAN”
“Get/SetNTPServer”, “EventNTPServerCnf”	“haveNTP”
“Get/SetOMPURL ” “EventOMPURLCnf”	“haveOMP”
“Get/SetPBXEnrolment” “EventPBXEnrolmentCnf”	“haveEnrolmentPBX” “haveProtFTP” “haveProtFTPS” “haveProtHTTP” “haveProtHTTPS” “haveProtSCP” “haveProtSFTP” “haveProtTFTP”
“Get/SetPPFirmwareURL” “Get/SetPPFirmwareUpdate” “GetPPFirmwareUpdateOverview” “GetPPFirmwareUpdateStatus” ”EventPPFirmwareURLCnf” ”EventPPFirmwareUpdateCnf” “EventPPFirmwareUpdateOverview” “EventPPFirmwareUpdateStatus”	“havePPFirmwareUpdate”
“Get/SetPPUser” ”EventPPUserCnf”	“haveExternalUserData” ”haveSIP” ”haveSOSNumberPP” ”haveUnboundDevices” “haveVoiceboxNumber” “haveConference” “haveBluetooth” “haveVideo” “haveHCM”
“Get/SetRegistrationTrafficShaping” “EventRegistrationTrafficShapingCnf”	”haveSIP”
“Get/SetRFP”	“havePagingAreas””haveWLAN”

“EventRFPCnf”	
“Get/SetRFBM” “EventRFBMCnf”	“haveRFBMCnf”
“Get/SetRTP” “EventRTPCnf”	“haveRTPPortBaseCnf” “haveSIP”
“Get/SetSNMP” “EventSNMPCnf”	“haveSNMP”
“Get/SetSpecialBranding”, “EventSpecialBrandingCnf”	“haveSpecialBranding”
“Get/SetSuplServ” “EventSuplServ”	“haveSIP”
“Get/SetSysVoiceboxNum” “EventSysVoiceboxNumCnf”	“haveVoiceboxNumber”
“Get/SetTimeZone” “EventTimeZoneCnf” “EventTimeZoneList”	“haveTZones”
“Get/SetTimeZoneDetails” “EventTimeZoneDetailsCnf”	“haveTZones”
“Get/SetUserDataServer” “EventUserDataServerCnf”	“haveExternalUserData” “haveUnboundDevices” “haveProtFTP” “haveProtFTPS” “haveProtHTTP” “haveProtHTTPS” “haveProtSCP” “haveProtSFTP” “haveProtTFTP”
“Get/SetUserDeviceSyncOMM” “EventUserDeviceSyncOMMCnf”	“haveUserDeviceSync”
“Get/SetPreserveUserDeviceRelation” “EventPreserveUserDeviceRelationCnf”	“haveExternalUserData” “haveUnboundDevices”
“Get/SetUserMonitoring” “EventUserMonitoringCnf”	“haveUMO”
“Get/SetWLANProfile” “EventWLANProfileCnf”	“haveWLAN”
“Get/SetWLANRegDomain” “EventWLANRegDomainCnf”	“haveWLAN”
“GetAutoDBBackupFileName” “EventAutoDBBackupFileNameCnf”	“haveAutoDB”
“GetDbTransferState”“EventDbTransferState”	“haveAutoDB”

“GetDigitTreatmentSummary”	“haveDigitTreatment” “haveDigitTreatmentSimple”
“Get/EventFreeConferenceChannels”	“haveConference”
“GetG729ChannelsForConference”	“haveConference”
“GetPPState” “EventPPState”	“haveSIP”
“Get/SetSoftwareImageURL” and “EventSoftwareImageURLCnf”	“haveSoftwareImageUR”
“Get/SetSyslogServer” and “EventSyslogServerCnf”	“haveOmmLogForward”
“GetTimeZoneList”	“haveTZones”
“CreateVideoDev” “DeleteVideoDev” “EventVideoDevCnf” “EventVideoDevLink” “GetVideoDev” “GetVideoDevLink” “SetVideoDev” “GetVideoDevSummary” “EventVideoDevSummary”	“haveVideo”
“GetWLANClients”	“haveWLAN”
“GetWLANRegDomainList”	“haveWLAN”
“ManualDBBackup” “ManualDBRestore”	“haveAutoDB” “haveProtFTP” “haveProtFTPS” “haveProtHTTP” “haveProtHTTPS” “haveProtSCP” “haveProtSFTP” “haveProtTFTP”
“ManualUserDataImport”	“haveExternalUserData” “haveUnboundDevices”
“Get/SetSARI” “EventSARICnf”	“haveSARI”
“PARKFromServer” “EventPARKFromServerResult”	“havePARKFile”
“PutFile”	“haveLicense”
“ResetTimeZoneDetails”	“haveTZones”
“Get/SetPPDev” “EventPPDevCnf”	“haveUnboundDevices” “haveBluetooth”

	"haveUserDeviceSync"
"GetLicensedCodecLines" "EventLicensedCodecLines"	"haveLicense" "haveLicensingCodec"
"Limits"	"haveWLAN"
"SystemRestart"	"haveRFPOMM"
"Open"	"haveAdditionalSettings" "haveAutoDB" "haveBluetooth" "haveConference" "haveConfigPython" "haveLDAP" "haveDECTRegDomain" "haveDigitTreatment" "haveDigitTreatmentSimple" "haveExternalUserData" "haveFACs" "haveEnrolmentPBX" "haveIMA" "haveLicensing" "haveLicensingCodec" "haveOmmLogForward " "haveOMP" "havePARKFile" "havePagingAreas" "haveProtFTP" "haveProtFTPS" "haveProtHTTP" "haveProtHTTPS" "haveProtSCP" "haveProtSFTP" "haveProtTFTP" "haveRFPMCnf" "haveRFPOMM" "haveRTPPortBaseCnf" "haveSIP" "haveSNMP" "haveSoftwareImageURL" "haveSOSNumberPP" "haveTZones" "haveUMO" "haveUnboundDevices" "haveVLAN" "haveVoiceboxNumber" "haveWLAN" "haveLocating" "havePPFirmwareUpdate" "haveIPParamCnf" "haveXML" "haveXMLDynamic" "haveXMLCorpDir"

5.2 Conditions For "OpenResp"-Parameters

The following table lists the constraints of all "OpenResp" capability parameters:

PARAMETER VALUE OF "OPENRESP"	Dependency
"haveAdditionalSettings=1"	If set, the requests/events "Get/SetAdditionalSettings" and "EventAdditionalSettingsCnf" are supported by the OMM.
"haveAutoDB=1"	If set, the requests/events "Get/SetAutoDBBackup", "GetAutoDBBackupFileName", "GetDbTransferState", "ManualDBBackup", "ManualDBRestore", "EventAutoDBBackupCnf", "EventAutoDBBackupFileNameCnf" and "EventDbTransferState" are supported by the OMM.
"haveBluetooth=1"	<p>If set:</p> <ol style="list-style-type: none"> The requests/events "Get/Set/Create/DeleteBluetoothBeacon", "Get/SetBluetoothGlobalSettings", "Get/SetBluetoothSensitivity", "Get/EventBluetoothClientStatistic", "Get/EventBluetoothBeaconSummary", "EventBluetoothGlobalSettings", "EventBluetoothSensitivity" and "EventBluetoothBeaconCnf" are supported by the OMM. Attributes "capBluetooth" and "ethAddr" of PPDevType are supported. Attribute "BTlocatable" of PPUserType is supported. Attribute "BTSensitivity" of PPUserType is supported.
"haveConfigPython=1"	<ol style="list-style-type: none"> If set, the requests „DBBackupToUSB“ and „DBRestoreFromUSB“ are supported by the OMM.
"haveConference=1"	<p>If set:</p> <ol style="list-style-type: none"> The requests/events "ConferenceConfirmation", "Create/DeleteConference", "Create/DeleteConferenceRoom", "AddUserToConference", "DeleteUserFromConference", "EventCreateConference", "EventAddUserToConference", "EventDeleteConference", "EventDeleteUserFromConference", "Get/EventReadyForConferencing", "EventConferenceRelease", "EventConferenceRequest", "EventConferenceRoomCnf", "ConferenceConfirmation", "Get/SetConferenceServerSIP", "EventConferenceServerSIPCnf", "GetG729ChannelsForConference", "GetFreeConferenceChannels", "EventFreeConferenceChannels" and "Get/SetConferenceRoom" are supported by the OMM. Attributes "conferenceServerURI" and "conferenceServerType" of PPUserType are supported.
"haveDECTPpSettings=1"	If set, the requests/events "Get/SetDECTPpSettings" and

	“EventDECTPpSettingsCnf” are supported by the OMM.
“haveDECTRegDomain=1”	If set, the requests/events “Get/SetDECTRegDomain” and “EventDECTRegDomainCnf” are supported by the OMM.
“haveDigitTreatment=1”	If set, the requests/events “Create/Delete/Get/SetDigitTreatment”, “GetDigitTreatmentSummary” and “EventDigitTreatmentCnf” are supported by the OMM.
“haveDigitTreatmentSimple=1”	If set, the requests/events “Create/Delete/Get/SetDigitTreatment”, “GetDigitTreatmentSummary” and “EventDigitTreatmentCnf” are supported by the OMM.
“haveLDAP=1”	If set, the requests/events “Create/Delete/Get/SetLDAP” and “EventLDAPCnf” are supported by the OMM.
“haveEnrolmentPBX=1”	If set, the requests/events “Get/SetPBXEnrolment” and “EventPBXEnrolmentCnf” are supported by the OMM.
“haveExternalUserData=1”	If set: <ol style="list-style-type: none"> 1. The requests/events “Get/SetUserDataServer” and “EventUserDataServerCnf” are supported by the OMM. 2. The requests/events “Get/SetPreserveUserDeviceRelation” and “EventPreserveUserDeviceRelationCnf” are supported by the OMM. 3. The PPUser attribute “external” cannot be read, set or notified.
“haveFACs=1”	If set, the requests/events “Get/SetFAC”, “SetFACList”, “Get/SetFACPrefix”, “EventFACPrefixCnf” and “EventFACCnf” are supported by the OMM.
“haveHCM=1”	If set: <ol style="list-style-type: none"> 1. The requests/events “Get/SetPpProfile” and “EventPpProfileCnf” are supported by the OMM. 2. . 3. The DECT phone device attributes “ppProfileCapability” and “ppDefaultProfileLoaded” are supported. 4. The <i>component</i> value “ppProfile” in <i>HealthType</i> is supported. 5. The <i>PPUserType</i> attributes “configurationDataLoaded”, “ppProfileId” and “ppData” are supported.
“haveIMA=1”	If set: <ol style="list-style-type: none"> 1. The requests/events “Get/SetIMA” and “EventIMACnf” are supported by the OMM. 2. IMA configuration file download (<i>PutFile</i> “:ima”) is supported
“haveIPParamCnf=1”	If set, the requests/events “Get/SetNetParams” and “EventNetParamsCnf” are supported by the OMM.
“haveLicensing=1”	If set:

	<ol style="list-style-type: none"> 3. The requests/events “GetLicense”, “GetLicensedCodeLines”, “EventLicensedCodeLines” and “EventLicenseCnf” are supported by the OMM 4. The requests/events “Get/SetLicenseServerList” and “EventLicenseServerListCnf” are supported by the OMM 5. The request “PutFile” supports the logical file “:license”. 6. The event “EventLicenseFile” is supported.
“haveLicensingCodec=1”	<p>If set:</p> <ol style="list-style-type: none"> 1. The requests/events “GetLicensedCodeLines” and “EventLicensedCodeLines” are supported by the OMM 2. Licensing for the number of G.729 connections in is supported in “GetLicense” and “EventLicenseCnf”.
“haveLocating=1”	<p>If not set:</p> <ol style="list-style-type: none"> 1. The permissions (see section 0) will never be set to “LocatingAlert”, “Locating” or “Alerting”. 2. Subscription (see 4.1.2) for “Locating”, “PositionTrack”, “AlarmTrigger” and “AlarmCallProgress” will not be supported. 3. Messages with priority “Emergency” or “Locating Alert” will be rejected. <p>If set:</p> <ol style="list-style-type: none"> 1. The requests/events “RequestPositionInfo”, “EventPositionHistory”, “EventPositionInfo”, “EventPositionRequest” and “EventPositionTrack” are supported by the OMM.
“haveNTP=1”	<ol style="list-style-type: none"> 1. If set, the requests/events “Get/SetNTPServer” and “EventNTPServerCnf” are supported by the OMM.
“haveOMP=1”	<p>If set, the requests/events “Get/SetOMPURL” and “EventOMPURLCnf” are supported by the OMM.</p>
“haveOmmLogForward=1”	<p>If set, the attribute “forward” in requests/events “Get/SetSyslogServer” and “EventSyslogServerCnf” is supported by the OMM.</p>
“havePARKFile=1”	<p>If set:</p> <ol style="list-style-type: none"> 1. The requests “PARKFromServer” is supported by the OMM. 2. The event “EventPARKFromServer” is supported by the OMM. 3. The file type “:park” in “GeFile” and “PutFile” is supported.
“havePagingAreas=1”	<p>If set:</p> <ol style="list-style-type: none"> 1. The requests/events “Get/SetDECTPagingAreaSize” and “EventDECTPagingAreaSizeCnf” are supported by the OMM. 2. The RFP attribute “pagingArea” can be read, set or notified.
“haveProtFTP=1” “haveProtFTPS=1” “haveProtHTTP=1” “haveProtHTTPS=1” “haveProtSCP=1”	<p>Depending on settings, the related protocols or <i>URLType</i> in requests/events “Get/SetAutoDBBackup”, “EventAutoDBBackupCnf”, “ManualDBBackup”, “ManualDBRestore”, “Get/SetUserDataServer”, “EventUserDataServerCnf”, “Get/SetPBXEnrolment”, “EventPBXEnrolmentCnf”, “CreateXMLApplication”,</p>

<p>“haveProtSFTP=1” “haveProtTFTP=1”</p>	<p>“Get/SetXMLApplication” and “EventXMLApplicationCnf” are supported by the OMM.</p>
<p>“havePPFirmwareUpdate=1”</p>	<p>If set, the requests/events “Get/SetPPFirmwareURL”, “Get/SetPPFirmwareUpdate”, “GetPPFirmwareUpdateOverview”, “GetPPFirmwareUpdateStatus”, “EventPPFirmwareURLCnf”, “EventPPFirmwareUpdateCnf”, “EventPPFirmwareUpdateOverview” and “EventPPFirmwareUpdateStatus” are supported by the OMM.</p>
<p>“haveRFPMCnf=1”</p>	<p>If set:</p> <ol style="list-style-type: none"> 1. The requests/events “Get/SetRFPM” and “EventRFPMCnf” are supported by the OMM. 2. The health state component type <i>dhcpServer</i> is supported.
<p>“haveRFPOMM=1”</p>	<p>If set:</p> <ol style="list-style-type: none"> 1. The attribute “resetToFactoryDefaults” in “SystemRestart” is supported by the OMM. 2. The RFP attribute “ntpServer” is supported by the OMM
<p>“haveRTPPortBaseCnf=1”</p>	<p>If set:</p> <ol style="list-style-type: none"> 1. The requests/events “Get/SetRTP” and “EventRTPCnf” are supported by the OMM. 2. The attributes “packetTime”, “silenceSupp” and “type” cannot be read, set or notified.
<p>“haveSARI=1”</p>	<p>If set:</p> <ol style="list-style-type: none"> 1. The requests/events “Get/SetSARI” and “EventSARICnf” are supported by the OMM
<p>“haveSIP=1”</p>	<p>If not set, the requests/events “Get/SetPPLLoginVariant” and “EventPPLLoginVariantCnf” are not supported by the OMM.</p> <p>If set:</p> <ol style="list-style-type: none"> 1. The requests/events “Get/SetRTP”, “Get/SetAdvancedSIP”, “Get/SetBasicSIP”, “Get/SetBackupSIP”, “Get/SetDTMF”, “Get/SetRegistrationTrafficShaping”, “Get/SetSuplServ”, “EventDTMFCnf”, “EventSuplServCnf”, “EventAdvancedSIPCnf”, “EventBasicSIPCnf”, “EventBackupSIPCnf”, “EventRegistrationTrafficShapingCnf”, “EventDTMFCnf” and “EventRTPCnf” are supported by the OMM. 2. The PPUUser attributes “addId”, “sipPw”, “sipAuthId”, “forwardState”, “forwardTime”, “forwardDest”, “holdRingBackTime” and “callWaitingDisabled” are supported by the OMM. 3. The RTP attributes “packetTime”, “silenceSupp” and “type” are supported by the OMM. 4. “Get/SetRegistrationTrafficShaping” and “EventRegistrationTrafficShapingCnf” are supported by the OMM. 5. The requests/events “Get/SetPPLLoginVariant” and “EventPPLLoginVariantCnf” are supported by the OMM, when also “haveUnboundDevices” is set. 6. The attributes “<i>regServerType</i>”, “<i>regServerAddr</i>” and

	<p>“<i>regServerPort</i>” of the request/event “<i>GetPPState</i>”/”<i>EventPPState</i>” are supported by the OMM.</p> <ol style="list-style-type: none"> The attributes “<i>fixedSipPort</i>” and “<i>calculatedSipPort</i>” in “<i>PPUserType</i>” and “<i>ConferenceRoomType</i>” are supported by the OMM. The attributes “<i>autoAnswer</i>”, “<i>MicrophoneMute</i>”, “<i>warningTone</i>” and “<i>allowBargeIn</i>” in “<i>PPUserType</i>” are supported by the OMM. The requests/events “<i>GetPortRangeSIP</i>”, “<i>SetPortRangeSIP</i>” and “<i>EventPortRangeSIPCnf</i>” are supported by the OMM. The requests/events “<i>GetIntercomCallHandlingSIP</i>”, “<i>SetIntercomCallHandlingSIP</i>” and “<i>EventIntercomCallHandlingCnf</i>” are supported by the OMM. The requests/events “<i>Get/SetSecureSIP</i>”, “<i>EventSecureSIPCnf</i>”, “<i>SetSecureSIPCertificate</i>”, “<i>GetSecureSIPCertificate</i>”, “<i>EventSecureSIPCertificateCnf</i>”, “<i>SetSecureSIPCertificateServerImport</i>”, “<i>GetSecureSIPCertificateServerImport</i>” and “<i>EventSecureSIPCertificateServerImportCnf</i>” are supported by the OMM.
“ <i>haveSNMP=1</i> ”	If set, the requests/events “ <i>Get/SetSNMP</i> ” and “ <i>EventSNMPCnf</i> ” are supported by the OMM.
“ <i>haveSoftwareImageURL=1</i> ”	If set, the requests/events “ <i>Get/SetSoftwareImageURL</i> ” and “ <i>EventSoftwareImageURLCnf</i> ” are supported by the OMM.
“ <i>haveSpecialBranding=1</i> ”	If set, the requests/events “ <i>Get/SetSpecialBranding</i> ” and “ <i>EventSpecialBrandingCnf</i> ” are supported by the OMM.
“ <i>haveSRTP=1</i> ”	<p>If set:</p> <ol style="list-style-type: none"> The attribute <i>sctp</i> in “<i>Create/Get/SetSite</i>” and “<i>EventSiteCnf</i>” is supported by the OMM. The attribute <i>sctpSiteNum</i> in “<i>Get/EventSiteSummary</i>” is supported by the OMM. The attribute <i>sctpProfile</i> in “<i>AddUserToConference</i>” and “<i>EventAddUserToConference</i>” is supported. The attributes <i>rxSecret</i> and <i>txSecret</i> in “<i>AddUserToConference</i>” are supported.
“ <i>haveSOSNumberPP=1</i> ”	If set the <i>PPUser</i> attributes “ <i>sosNum</i> ” and “ <i>manDownNum</i> ” are supported by the OMM.
“ <i>haveTZones=1</i> ”	If set, the requests/events “ <i>Get/SetTimeZone</i> ”, “ <i>Get/SetTimeZoneDetails</i> ”, “ <i>GetTimeZoneList</i> ”, “ <i>RequestResetTimeZoneDetails</i> ”, “ <i>EventTimeZoneCnf</i> ”, “ <i>EventTimeZoneList</i> ” and “ <i>EventTimeZoneDetailsCnf</i> ” are supported by the OMM.
“ <i>haveUnboundDevices=1</i> ”	<p>If not set, the requests/events “<i>Get/SetPPLLoginVariant</i>” and “<i>EventPPLLoginVariantCnf</i>” are not supported by the OMM.</p> <p>If set:</p> <ol style="list-style-type: none"> The requests/events “<i>Get/SetDevAutoCreate</i>”,

	<p>“Get/SetPPLoginVariant “CreatePPDev”, “CreatePPUser”, “EventPPLoginVariantCnf”, “EventUserDataImport”, “ManualUserDataImport” and “EventDevAutoCreateCnf” are supported by the OMM.</p> <ol style="list-style-type: none"> The PPUser attribute “pin” is supported by the OMM. The FAC attribute “fac” does support the values “UserLogin”, and “UserLogout” The requests/events “Get/SetPPLoginVariant” and “EventPPLoginVariantCnf” are supported by the OMM, when also “haveSIP”. The PPUser attribute “keepLocalPB” is supported by the OMM.
“haveUMO=1”	<p>If set:</p> <ol style="list-style-type: none"> The requests/events “Get/SetUserMonitoring” and “EventUserMonitoringCnf” are supported by the OMM The requests/events “Get/SetUserMonitoring” and “EventUserMonitoringCnf” support the user data attributes “passiveMonitoring”, “activeMonitoring”, “HAS”, “HSS”, “HRS”, “HCS”, “SRS”, “SCS”, “CDS”, “HBS” and “BTS”. User monitoring health states are supported. User monitoring alarm triggers are supported
“haveUserDeviceSync=1”	<p>If set:</p> <ol style="list-style-type: none"> The requests/events “Get/SetUserDeviceSyncOMM” and “EventUserDeviceSyncOMMCnf” are supported by the OMM The attribute “haveUserDeviceSync” in “PPDevType” is supported by the OMM
“haveVideo=1”	<p>If set:</p> <ol style="list-style-type: none"> The requests/events “CreateVideoDev”, “DeleteVideoDev”, “EventVideoDevCnf”, “EventVideoDevLink”, “GetVideoDev”, “GetVideoDevSummary”, “EventVideoDevSummary”, “GetVideoDevLink” and “SetVideoDev” are supported by the OMM. The attribute “allowVideoStream” is supported in the PPUser data set (PPUserType). The attribute “videoStreaming” is supported in the SiteType. The attribute “” is supported in the “Get/EventSiteSummary”.
“haveVLAN=1”	<p>If set, the requests/events “Get/SetNetParams” and “EventNetParamsCnf” support the attributes “voiceEthPrio” and “sigEthPrio”.</p>
“haveVoiceboxNumber=1”	<p>If set, the requests/events “Get/SetSysVoiceboxNum”, “EventSysVoiceboxNumCnf” and the attributes “voiceboxNum” in <i>PPUserType</i> are supported by the OMM.</p>
“haveWLAN=1”	<p>If set:</p> <ol style="list-style-type: none"> The requests/events “Get/SetWLANProfile”, “Get/SetWLANRegDomain”, “RequestGetACLEnt”,

	<p>“RequestSetACLEntry “, “RequestGetWLANClients “, “RequestGetWLANRegDomainList “, “Create/DeleteACLEntry”, “Create/DeleteWLANProfile.cpp”, “EventACLConf”, “EventWLANClient”, “EventWLANProfileConf” and “EventWLANRegDomainConf” are supported by the OMM.</p> <p>2. The RFP attributes “wlanProfile”, “wlanAntennaDiv”, “wlanAntenna”, “wlanChannel”, “wlanPower”, “wlanRunning”, “wlanLinkNok”, “wlanChannelUsed”, “wlanPowerUsed” and “wlanHighThroughputTypeUsed” are supported by the OMM.</p> <p>3. The attributes “wlanClients” and “wlanProfiles” in request “Limits are supported.</p>
“haveXML=1”	If set, the requests/events “Get/SetXMLApplication”, “Create/DeleteXMLApplication” and “EventXMLApplicationConf” are supported.
“haveXML=1”and “haveXMLDynamic=1”	If set, the requests/events “Create/Delete/Get/SetXMLApplication” and “EventXMLApplicationConf” are supported.
“haveXMLCorpDir=1”	If set, the attribute <i>corpDirOrder</i> and the attribute setting <i>type=“CorpDir”</i> of requests/events “Create/Delete/Get/SetXMLApplication” and “EventXMLApplicationConf” are supported.

5.3 OMM Version History

The following table shows the history of the OMM versions.

OMM AXI Specification version	Requests/Responses or Event Versions	Comment
3.11	All Versions start with version „1.0.0”	
3.12	CreateRFP „1.1.0” CreateWLANProfile „2.0.0” EventLicenseConf „1.1.0” EventRFPConf „1.1.0” EventRFPConnectAttempt „1.1.0” EventRFPDetails „1.1.0” EventRFPState „1.1.0” EventRTPConf „2.0.0” EventWLANProfileConf „2.0.0” GetLicense „1.1.0” GetRFP „1.1.0” GetRFPConf „1.1.0” GetRFPConfList „1.1.0” GetRTP „2.0.0” GetWLANProfile „2.0.0” SetRFP „1.1.0” SetRFPConf „1.1.0” SetRTP „2.0.0” SetWLANProfile „2.0.0”	
3.13	GetVersions „2.0.0” SetDECTRegDomain „1.1.0”	

	<i>SetWLANRegDomain „1.1.0”</i>	
3.14	<i>CreateAlarmTrigger „1.0.1”</i> <i>CreatePPUser „1.1.0”</i> <i>CreateFixedPP „1.1.0”</i> <i>CreateRFP „1.2.0”</i> <i>DeleteAlarmTrigger „1.0.1”</i> <i>EventAlarmTrigger „1.0.1”</i> <i>EventHealthState „1.0.1”</i> <i>EventMessageSend „1.0.2”</i> <i>EventPPUserCnf „1.1.0”</i> <i>GetHealthState „1.0.1”</i> <i>GetPPState „1.0.1”</i> <i>GetPPUser „1.1.0”</i> <i>GetXMLApplication „1.0.1”</i> <i>SendMessage „1.0.1”</i> <i>SetAccount „1.0.1”</i> <i>SetAlarmTrigger „1.0.1”</i> <i>SetPPUser „1.1.0”</i> <i>Subscribe „1.0.1”</i>	New alarm trigger G729ABORT New DECT phone languages New DECT phone languages New error codes/WlanRegDomainInvalid New alarm trigger G729ABORT New alarm trigger G729ABORT New health state reason code New optional attribute alwaysIndirect and internal usage New DECT phone languages New health state reason code Bugfix New DECT phone languages Bugfix Internal usage Bugfix New alarm trigger G729ABORT New DECT phone languages Bugfix
4.1	<i>AddUserToConference „1.0.0”</i> <i>ChangeUserInConference „1.0.0”</i> <i>ConferenceConfirmation „1.0.0”</i> <i>CreateAccount „1.0.1”</i> <i>CreateAlarmTrigger „1.0.2”</i> <i>CreateBluetoothBeacon „1.0.0”</i> <i>CreateConference „1.0.0”</i> <i>CreateConferenceRoom „1.0.0”</i> <i>CreateFixedPP „1.1.1”</i> <i>CreateLDAP „1.0.1”</i> <i>CreatePPDev „1.0.1”</i> <i>CreatePPUser „1.1.1”</i> <i>CreateRFP „1.3.0”</i> <i>CreateVideoDev „1.0.0”</i> <i>CreateXMLApplication „2.0.0”</i> <i>DBBackupToUSB „1.0.0”</i> <i>DBRestoreFromUSB „1.0.0”</i> <i>DeleteAlarmTrigger „1.0.2”</i> <i>CreateBluetoothBeacon „1.0.0”</i> <i>DeleteConference „1.0.0”</i> <i>DeleteConferenceRoom „1.0.0”</i> <i>DeleteUserFromConference „1.0.0”</i> <i>DeleteVideoDev „1.0.0”</i> <i>DeleteXMLApplication „2.0.0”</i> <i>EventAccountCnf „1.0.1”</i> <i>EventAccountSummary „1.0.0”</i> <i>EventAddUserToConference „1.0.0”</i> <i>EventAdvancedSIPCnf „2.0.0”</i> <i>EventAlarmTrigger „1.0.2”</i> <i>EventAutoDBBackupCnf „1.0.1”</i> <i>EventBackupSIPCnf „1.0.0”</i>	

<p> <i>EventBluetoothClientStatistic „1.0.0”</i> <i>EventBluetoothBeaconCnf „1.0.0”</i> <i>EventBluetoothSensitivityCnf „1.0.0”</i> <i>EventChangeUserInConference „1.0.0”</i> <i>EventConference „1.0.0”</i> <i>EventConferenceRequest „1.0.0”</i> <i>EventConferenceRoomCnf „1.0.0”</i> <i>EventConferenceServerSIP Cnf „1.0.0”</i> <i>EventDeleteConference „1.0.0”</i> <i>EventCreateConference „1.0.0”</i> <i>EventDeleteUserFromConference „1.0.0”</i> <i>EventFlashMemUsage“0.0.0”</i> <i>EventFreeConferenceChannels „1.0.0”</i> <i>EventHealthState „1.1.0”</i> <i>EventMessageSend „1.0.3”</i> <i>EventPermissionChange „1.0.1”</i> <i>EventPBXEnrolmentCnf „1.0.1”</i> <i>EventPPDevCnf „1.0.1”</i> <i>EventPPDevSummary „1.0.0”</i> <i>EventPPFirmwareUpdateCnf „1.0.1”</i> <i>EventPPFirmwareUpdateStatus „1.0.1”</i> <i>EventPPState „1.1.0”</i> <i>EventPPUserSummary „1.0.1”</i> <i>EventPPUserCnf „1.1.1”</i> <i>EventReadyForConferencing „1.0.0”</i> <i>EventRFPCnf „1.2.0”</i> <i>EventRFPSummary „1.0.1”</i> <i>EventRTPCnf „3.0.0”</i> <i>EventRTPConferenceStreamChg „1.0.0”</i> <i>EventSiteSummary „1.0.0”</i> <i>EventSoftwareUpdateCnf „1.0.0”</i> <i>EventSuplServCnf „1.0.1”</i> <i>EventToneSchemeCnf „1.0.0”</i> <i>EventUserDataImport „1.0.0”</i> <i>EventUserDataServerCnf „1.0.1”</i> <i>EventUserMonitoringCnf „1.0.0”</i> <i>EventVideoDevCnf „1.0.0”</i> <i>EventVideoDevLink „1.0.0”</i> <i>EventWLANClient „1.0.1”</i> <i>EventXMLApplicationCnf „2.0.0”</i> <i>GetAccount „1.0.1”</i> <i>GetAccountSummary „1.0.1”</i> <i>GetAdvancedSIP „2.0.0”</i> <i>GetAlarmTrigger „1.1.0”</i> <i>GetBackupSIP „1.0.0”</i> <i>GetBluetoothClientStatistic „1.0.0”</i> <i>GetBluetoothBeacon „1.0.0”</i> <i>GetBluetoothSensitivity „1.0.0”</i> <i>GetConferenceRoom „1.0.0”</i> <i>GetConferenceServerSIP „1.0.0”</i> </p>	
---	--

<p> <i>Get G729ChannelsForConference „1.0.0”</i> <i>Get FACdeleted</i> <i>Get FACList „1.0.0”</i> <i>GetFreeConferenceChannels „1.0.0”</i> <i>GetG729ChannelsForConference „1.0.0”</i> <i>GetHealthState „1.1.0”</i> <i>GetPPDevSummary „1.0.1”</i> <i>GetPPFirmwareUpdateOverview „1.0.1”</i> <i>GetPPFirmwareUpdate „1.0.1”</i> <i>GetPPFirmwareUpdateStatus „1.0.1”</i> <i>GetPPUserSummary „1.0.1”</i> <i>GetSoftwareUpdate „1.0.0”</i> <i>GetPBXEnrolment „1.0.1”</i> <i>GetPPDev „1.0.1”</i> <i>GetPPState „1.1.0”</i> <i>GetPPUser „1.2.0”</i> <i>GetReadyForConferencing „1.0.0”</i> <i>GetRFP „1.2.0”</i> <i>GetRFPCaptureList „1.1.1”</i> <i>GetRFPSummary „1.0.1”</i> <i>GetRTP „3.0.0”</i> <i>GetSiteSummary „1.0.1”</i> <i>GetSuplServ „1.0.1”</i> <i>GetTimeZoneDetails „1.0.1”</i> <i>GetToneSchemeCnf „1.0.0”</i> <i>GetVersions „2.1.0”</i> <i>GetVideoDev „1.0.0”</i> <i>GetVideoDevLink „1.0.0”</i> <i>GetUserDataServer „1.0.1”</i> <i>GetUserMonitoring „1.0.0”</i> <i>GetXMLApplication „2.0.0”</i> <i>Limits „1.1.0”</i> <i>ManualDBBackup „1.0.1”</i> <i>ManualDBRestore „1.0.1”</i> <i>ManualUserDataImport „1.0.0”</i> <i>Open „1.0.2”</i> <i>SendMessage „1.0.2”</i> <i>SetAccount „1.0.2”</i> <i>SetAdvancedSIP „2.0.0”</i> <i>SetAutoDBBackup „1.0.1”</i> <i>SetConferenceRoom „1.0.0”</i> <i>SetConferenceServerSIP „1.0.0”</i> <i>SetAlarmTrigger „1.0.2”</i> <i>SetBackupSIP „1.0.0”</i> <i>SetBluetoothBeacon „1.0.0”</i> <i>SetBluetoothSensitivity „1.0.0”</i> <i>Set FACList „1.0.0”</i> <i>SetLDAP „1.0.1”</i> <i>SetPBXEnrolment „1.0.1”</i> <i>SetPPDev „1.0.1”</i> <i>SetPPUser „1.1.1”</i> </p>	
--	--

	<p><i>SetPPUserDevRelation „1.0.0”</i> <i>SetRFP „1.2.0”</i> <i>SetRTP „3.0.0”</i> <i>SetRTPConferenceStreamChg „1.0.0”</i> <i>SetSuplServ „1.0.0”</i> <i>SetTimeZoneDetails „1.0.1”</i> <i>SetToneScheme „1.0.0”</i> <i>SetUserDataServer „1.0.1”</i> <i>SetUserMonitoring „1.0.0”</i> <i>SetVideoDev „1.0.0”</i> <i>SetXMLApplication „2.0.0”</i> <i>SoftwareUpdate „1.0.1”</i> <i>Subscribe „3.0.0”</i></p>	
<p>4.2 (from not released version!)</p>	<p><i>AddUserToConference „2.0.0”</i> <i>ChangeUserInConference „2.0.0”</i> <i>EventAddUserToConference „2.0.0”</i> <i>EventChangeUserInConference „2.0.0”</i> <i>EventConferenceRelease „1.0.0”</i></p> <p><i>CreateBluetoothBeacon „1.0.1”</i> <i>EventBluetoothBeaconCnf „1.0.1”</i> <i>GetBluetoothBeacon „1.0.1”</i> <i>SetBluetoothBeacon „1.0.1”</i></p> <p><i>EventAutoDBState „1.0.1”</i> <i>GetAutoDBState „1.0.1”</i></p> <p><i>EventDbTransferState „1.0.0”</i> <i>GetDbTransferState „1.0.0”</i></p>	<p>Attribute changes, new <i>ConferenceDataType</i> data type.</p> <p>New</p> <p>New optional read only attribute failed</p> <p>New attribute Values</p> <p>New</p>
<p>5.1</p>	<p><i>EventAutoDBState „1.0.1”</i> <i>GetAutoDBState „1.0.1”</i></p> <p><i>CreatePPDevProfile „1.0.0”</i> <i>DeletePPDevProfile „1.0.0”</i> <i>EventBluetoothBeaconSummary „1.0.0”</i> <i>EventBluetoothGlobalSettingsCnf „1.0.0”</i> <i>EventDbTransferState „1.0.1”</i> <i>EventDECTPpSettingsCnf „1.0.0”</i> <i>EventDefaultDevProfileCnf „1.0.0”</i> <i>EventLicenseFile „1.0.0”</i> <i>EventLicenseServerListCnf „1.0.0”</i> <i>EventOMPCnf „1.0.0”</i> <i>EventPPDevProfileCnf „1.0.0”</i> <i>EventRFPipQuality „1.0.0”</i> <i>EventRFPMediaStreamQuality „1.0.0”</i> <i>EventRFPsyncQuality „1.0.0”</i> <i>EventSecureSIPCertificateCnf „1.0.0”</i> <i>EventSecureSIPCnf „1.0.0”</i> <i>EventTimeZoneList „1.0.0”</i> <i>EventVideoDevSummary „1.0.0”</i> <i>GetAdditionalSettings „1.0.0”</i></p>	<p>Deleted:</p> <p>New:</p>

	<p> <i>GetBluetoothBeaconSummary „1.0.0”</i> <i>GetBluetoothGlobalSettings „1.0.0”</i> <i>GetDbTransferState „1.0.0”</i> <i>GetDECTPpSettings „1.0.0”</i> <i>GetDefaultDevProfile „1.0.0”</i> <i>GetLicenseServerList „1.0.0”</i> <i>GetOMP „1.0.0”</i> <i>GetPPDevProfile „1.0.0”</i> <i>GetRFPpQuality „1.0.0”</i> <i>GetRFPMediaStreamQuality „1.0.0”</i> <i>GetRFPsSyncQuality „1.0.0”</i> <i>GetSecureSIP „1.0.0”</i> <i>GetSecureSIPCertificate „1.0.0”</i> <i>GetVideoDevSummary „1.0.0”</i> <i>ResetRFPMediaStreamQuality „1.0.0”</i> <i>SetACLEntry „1.0.0”</i> <i>SetAdditionalSettings „1.0.0”</i> <i>SetBluetoothGlobalSettings „1.0.0”</i> <i>SetDECTPpSettings „1.0.0”</i> <i>SetDefaultDevProfile „1.0.0”</i> <i>SetLicenseServerList „1.0.0”</i> <i>SetOMP „1.0.0”</i> <i>SetPPDevProfile „1.0.0”</i> <i>SetSecureSIP „1.0.0”</i> <i>SetSecureSIPCertificate „1.0.0”</i> <i>UpdateIMAConfigFile „1.0.0”</i> </p> <p> <i>AddUserToConference „2.0.1”</i> <i>CreateACLEntry „1.0.2”</i> <i>CreateAlarmTrigger „1.0.3”</i> <i>CreateBluetoothBeacon „2.0.0”</i> <i>CreateFixedPP „1.1.2”</i> <i>CreatePPDev „1.0.2”</i> <i>CreatePPUser „1.1.2”</i> <i>CreateRFP „1.3.2”</i> <i>CreateSite „1.0.1”</i> <i>CreateVideoDev „2.0.0”</i> <i>CreateWLANProfile „2.0.2”</i> <i>DeleteACLEntry „1.0.2”</i> <i>DeleteRFPcaptureList „1.0.1”</i> <i>DeleteRFPcaptureListElem „1.0.1”</i> <i>EventACL Cnf „1.0.2”</i> <i>EventAddUserToConference „2.0.1”</i> <i>EventAdvancedSIPCnf „2.0.1”</i> <i>EventAlarmTrigger „1.0.3”</i> <i>EventBasicSIPCnf „2.0.0”</i> <i>EventBluetoothBeaconCnf „2.0.0”</i> </p> <p> <i>EventHealthState „1.1.3”</i> <i>EventLicenseCnf „2.0.0”</i> <i>EventMessageSend „1.0.5”</i> </p>	<p>Changed:</p>
--	--	-----------------

	<p> <i>EventPPDevCnf „1.0.2"</i> <i>EventPPUserCnf „1.1.2"</i> <i>EventRFPCapture „1.0.2"</i> <i>EventRFPCConnectAttempt „1.1.1"</i> <i>EventRFPCnf „2.0.0"</i> <i>EventRFPCDetails „1.1.3"</i> <i>EventRFPCState „1.1.3"</i> <i>EventRFPCSummary „1.0.3"</i> <i>EventSiteCnf „1.0.1"</i> <i>EventSiteSummary „1.0.1"</i> <i>EventVideoDevCnf „2.0.0"</i> <i>EventWLANProfileCnf „2.0.2"</i> <i>GetACLEntry „1.0.1"</i> <i>GetAdvancedSIP „2.0.1"</i> <i>GetBasicSIP „2.0.0"</i> <i>GetBluetoothBeacon „2.0.0"</i> <i>GetEventLogBuffer „1.0.1"</i> <i>GetHealthState „1.1.3"</i> <i>GetLicense „2.0.0"</i> <i>GetPPDev „1.0.2"</i> <i>GetPPUser „1.2.1"</i> <i>GetRFP „2.0.0"</i> <i>GetRFPCaptureList „1.1.3"</i> <i>GetRFPCSummary „1.0.3"</i> <i>GetSite „1.0.1"</i> <i>GetSiteSummary „1.0.2"</i> <i>GetTimeZoneList „2.0.0"</i> <i>GetVideoDev „2.0.0"</i> <i>GetWLANProfile „2.0.2"</i> <i>Limits „2.0.0"</i> <i>Open „1.0.6"</i> <i>SendMessage „1.0.3"</i> <i>SetAdvancedSIP „2.0.1"</i> <i>SetAlarmTrigger „1.0.3"</i> <i>SetBluetoothBeacon „2.0.0"</i> <i>SetDECTEncryption „1.0.1"</i> <i>SetDECTSubscriptionMode „1.0.1"</i> <i>SetPPDev „1.0.2"</i> <i>SetPPUser „1.1.2"</i> <i>SetRFP „1.2.2"</i> <i>SetSite „1.0.1"</i> <i>SetTimeZoneDetails „1.0.2"</i> <i>SetVideoDev „2.0.0"</i> <i>SetWLANProfile „2.0.2"</i> <i>SoftwareUpdate „1.0.2"</i> </p>	
5.2	<p> <i>CreateVideoDev „2.1.0"</i> <i>EventVideoDevCnf „2.1.0"</i> <i>GetVideoDev „2.1.0"</i> <i>SetVideoDev „2.1.0"</i> </p> <p> <i>EventSecureSIPCertificateServerImportCnf „1.0.0"</i> </p>	<p>Bugfix:</p> <p>New:</p>

	<p><i>GetSecureSIPCertificateServerImport „1.0.0”</i> <i>SetSecureSIPCertificateServerImport „1.0.0”</i> <i>EventDECTPpSettingsCnf „2.0.0”</i> <i>EventHealthState „1.1.4”</i> <i>EventPPUserSummary „1.0.2”</i> <i>GetDECTPpSettings „2.0.0”</i> <i>GetHealthState „1.1.4”</i> <i>GetPPUserSummary „1.0.2”</i> <i>GetVersions „2.1.1”</i> <i>SetDECTPpSettings „2.0.0”</i></p>	<p>Changed:</p>
5.3	<p><i>CancelMessage „1.0.1”</i> <i>CreateAlarmTrigger „1.0.5”</i> <i>DeleteMessage „1.0.1”</i> <i>SendMessage „1.0.5”</i> <i>SetAdvancedSIP „2.0.3”</i> <i>EventSuplServCnf „1.0.4”</i> <i>GetSuplServ „1.0.4”</i> <i>Limits „2.0.1”</i> <i>SetSuplServ „1.0.2”</i></p>	<p>Bugfix:</p> <p>New optional attributes New optional attributes New optional attributes New optional attributes</p>
6.0	<p><i>CreatePPDevProfile</i> <i>DeletePPDevProfile</i> <i>EventAutoDBRestoreCnf</i> <i>EventDefaultDevProfileCnf</i> <i>EventOMPCnf</i> <i>EventPPDevProfileCnf</i> <i>GetAutoDBRestore</i> <i>GetDefaultDevProfile</i> <i>GetOMP</i> <i>GetPPDevProfile</i> <i>SetAutoDBRestore</i> <i>SetDefaultDevProfile</i> <i>SetOMP</i> <i>SetPPDevProfile</i> <i>UpdateIMAConfigFile</i></p> <p><i>EventConfigURLCnf „1.0.0”</i> <i>EventCoreDumpURLCnf „1.0.0”</i> <i>EventIntercomCallHandlingSIPCnf „1.0.0”</i> <i>EventNTPServerCnf „1.0.0”</i> <i>EventOMMCertificateCnf „1.0.0”</i> <i>EventOMPURLCnf „1.0.0”</i> <i>EventPARKFromServerResult „1.0.0”</i> <i>EventPortRangeSIPCnf „1.0.0”</i> <i>EventPPFirmwareURLCnf „1.0.0”</i> <i>EventPpProfileCnf „1.0.0”</i> <i>EventPreserveUserDeviceRelationCnf „1.0.0”</i> <i>EventRestrictedSubscriptionDurationCnf „1.0.0”</i> <i>EventSoftwareImageURLCnf „1.0.0”</i> <i>EventSpecialBrandingCnf „1.0.0”</i> <i>EventSystemCredentialsCnf „1.0.0”</i> <i>EventSystemProvUpdTrigCnf „1.0.0”</i></p>	<p>Deleted:</p> <p>New:</p>

<p> <i>EventUsedConfigURL „1.0.0"</i> <i>GetConfigURL „1.0.0"</i> <i>GetCoreDumpURL „1.0.0"</i> <i>GetIntercomCallHandlingSIP „1.0.0"</i> <i>GetNTPServer „1.0.0"</i> <i>GetOMMCertificate „1.0.0"</i> <i>GetOMPURL „1.0.0"</i> <i>GetPortRangeSIP „1.0.0"</i> <i>GetPPFirmwareURL „1.0.0"</i> <i>GetPpProfile „1.0.0"</i> <i>GetPreserveUserDeviceRelation „1.0.0"</i> <i>GetRestrictedSubscriptionDuration „1.0.0"</i> <i>GetSoftwareImageURL „1.0.0"</i> <i>GetSpecialBranding „1.0.0"</i> <i>GetSystemCredentials „1.0.0"</i> <i>GetSystemProvUpdTrig „1.0.0"</i> <i>GetUsedConfigURL „1.0.0"</i> <i>PARKFromServer „1.0.0"</i> <i>SetConfigURL „1.0.0"</i> <i>SetCoreDumpURL „1.0.0"</i> <i>SetIntercomCallHandlingSIP „1.0.0"</i> <i>SetNTPServer „1.0.0"</i> <i>SetOMMCertificate „1.0.0"</i> <i>SetOMPURL „1.0.0"</i> <i>SetPortRangeSIP „1.0.0"</i> <i>SetPPFirmwareURL „1.0.0"</i> <i>SetPpProfile „1.0.0"</i> <i>SetPreserveUserDeviceRelation „1.0.0"</i> <i>SetRestrictedSubscriptionDuration „1.0.0"</i> <i>SetSoftwareImageURL „1.0.0"</i> <i>SetSpecialBranding „1.0.0"</i> <i>SetSystemCredentials „1.0.0"</i> <i>SetSystemProvUpdTrig „1.0.0"</i> <i>SystemRestart „1.0.1"</i> <i>CreateFixedPP „1.1.3"</i> <i>CreateConferenceRoom „1.0.1"</i> <i>CreateXMLApplication „3.0.0"</i> <i>CreatePPUser „1.1.4"</i> <i>DeletePPUser „1.0.1"</i> <i>EventAdvancedSIPCnf „3.0.0"</i> <i>EventAlarmTrigger „1.0.4"</i> <i>EventAutoDBBackupCnf „2.0.0"</i> <i>EventBasicSIPCnf „3.0.0"</i> <i>EventConferenceRoomCnf „1.0.1"</i> <i>EventDbTransferState „1.0.2"</i> <i>EventFACCnf „1.0.1"</i> <i>EventHealthState „1.1.5"</i> <i>EventIMACnf „2.0.0"</i> <i>EventPBXEnrolmentCnf „2.0.0"</i> <i>EventPPDevCnf „1.0.3"</i> <i>EventPPState „1.1.2"</i> </p>	<p>Changed:</p>
---	-----------------

<p> <i>EventPPUserCnf „1.1.3"</i> <i>EventRegistrationTrafficShapingCnf „2.0.0"</i> <i>EventSecureSIPCnf „2.0.0"</i> <i>EventSecureSIPCertificateCnf „1.1.0"</i> <i>EventSecureSIPCertificateServerImportCnf „2.0.0"</i> <i>EventSuplServCnf „1.0.5"</i> <i>EventSyslogServerCnf „1.0.1"</i> <i>EventUserDataServerCnf „2.0.0"</i> <i>EventXMLApplicationCnf „3.0.0"</i> <i>GetAdvancedSIP „3.0.0"</i> <i>GetAutoDBBackup „2.0.0"</i> <i>GetBasicSIP „3.0.0"</i> <i>GetConferenceRoom „1.0.1"</i> <i>GetDbTransferState „1.0.1"</i> <i>GetFACList „1.0.1"</i> <i>GetFile „1.0.1"</i> <i>GetHealthState „1.1.5"</i> <i>GetIMA „2.0.0"</i> <i>GetPBXEnrolment „2.0.0"</i> <i>GetPPDev „1.0.3"</i> <i>GetPPState „1.1.1"</i> <i>GetPPUser „1.2.2"</i> <i>GetRegistrationTrafficShaping „2.0.0"</i> <i>GetSecureSIP „2.0.0"</i> <i>GetSecureSIPCertificate „1.1.0"</i> <i>GetSecureSIPCertificateServerImport „2.0.0"</i> <i>GetSuplServ „1.0.5"</i> <i>GetSyslogServer „1.0.1"</i> <i>GetUserDataServer „2.0.0"</i> <i>GetXMLApplication „3.0.0"</i> <i>Limits „2.0.2"</i> <i>ManualDBBackup „2.0.0"</i> <i>ManualDBRestore „2.0.0"</i> <i>Open „1.0.7"</i> <i>PutFile „1.1.0"</i> <i>SetAdvancedSIP „3.0.0"</i> <i>SetAutoDBBackup „2.0.0"</i> <i>SetBasicSIP „2.0.0"</i> <i>SetConferenceRoom „1.0.1"</i> <i>SetFAC „1.0.1"</i> <i>SetFACList „1.0.1"</i> <i>SetIMA „2.0.0"</i> <i>SetPBXEnrolment „2.0.0"</i> <i>SetPPDev „1.0.3"</i> <i>SetPPLginVariant „1.0.1"</i> <i>SetPPUser „1.1.3"</i> <i>SetRegistrationTrafficShaping „2.0.0"</i> <i>SetSecureSIP „2.0.0"</i> <i>SetSecureSIPCertificate „1.1.0"</i> <i>SetSecureSIPCertificateServerImport „2.0.0"</i> <i>SetSuplServ „1.0.3"</i> </p>	
---	--

	<p><i>SetSyslogServer „1.0.1"</i> <i>SetUserDataServer „2.0.0"</i> <i>SetWLANProfile „2.0.3"</i> <i>SetXMLApplication „3.0.0"</i> <i>Subscribe „3.1.0"</i></p>	
6.0.2	<p><i>AddUserToConference „1.0.0"</i> <i>Event RemoteSystemDumpCnf „1.0.0"</i> <i>EventRFPMCnf „1.0.0"</i> <i>EventSecureOMMCertificateServerImportCnf „1.0.2"</i> <i>EventSecurePROVCertificateServerImportCnf „1.0.2"</i> <i>EventSecureSIPCertificateServerImportCnf „2.0.2"</i> <i>Get RemoteSystemDump „1.0.0"</i> <i>GetRFPM „1.0.0"</i> <i>GetSecureOMMCertificateServerImport „1.0.2"</i> <i>GetSecurePROVCertificateServerImport „1.0.2"</i> <i>GetSecureSIPCertificateServerImport „2.0.2"</i> <i>Set RemoteSystemDump „1.0.0"</i> <i>SetRFPM „1.0.0"</i> <i>SetSecureOMMCertificateServerImport „1.0.2"</i> <i>SetSecurePROVCertificateServerImport „1.0.2"</i> <i>SetSecureSIPCertificateServerImport „2.0.2"</i></p> <p><i>CreateAlarmTrigger "1.1.6"</i> <i>CreateFixedPPUser "1.1.4"</i> <i>CreatePPUser "1.1.5"</i> <i>CreateRFP "2.0.0"</i> <i>CreateWLANProfile "2.0.3"</i> <i>CreateXMLApplication "3.0.1"</i> <i>DeleteRFP "1.0.1"</i> <i>EventAlarmTrigger "1.1.5"</i> <i>EventAutoDBBackupCnf "2.0.1"</i> <i>EventCoreDumpURLCnf "1.0.1"</i> <i>EventHealthState"1.1.6"</i> <i>EventIMACnf "2.0.1"</i> <i>EventLicenseCnf "2.0.1"</i> <i>EventOMMCertificateCnf "1.0.1"</i> <i>EventOMPURLCnf "1.0.1"</i> <i>EventPBXEnrolmentCnf "2.0.1"</i> <i>EventPermissionChange "1.0.1"</i> <i>EventPPFirmwareURLCnf "1.0.1"</i> <i>EventRegistrationTrafficShapingCnf "2.0.1"</i> <i>EventRemoteSystemDumpCnf "1.0.1"</i> <i>EventRFPCnf "3.0.0"</i> <i>EventSecureOMMCertificateServerImportCnf "1.0.3"</i> <i>EventSecurePROVCertificateServerImportCnf "1.0.3"</i> <i>EventSecureSIPCertificateServerImportCnf "2.0.3"</i> <i>EventSoftwareImageURLCnf "1.0.1"</i> <i>EventSpecialBrandingCnf "1.0.1"</i> <i>EventUserDataServerCnf "2.0.1"</i> <i>EventXMLApplicationCnf "3.0.1"</i> <i>GetAlarmTrigger "1.1.2"</i></p>	<p>New:</p> <p>BugFix change:</p>

	<p> <i>GetAutoDBBackup</i> "2.0.1" <i>GetCoreDumpURL</i> "1.0.1" <i>GetHealthState</i>"1.1.6" <i>GetIMA</i> "2.0.1" <i>GetLicense</i> "2.0.1" <i>GetOMMCertificate</i> "1.0.1" <i>GetOMPURL</i> "1.0.1" <i>GetPBXEnrolment</i> "2.0.1" <i>GetPPFirmwareURL</i> "1.0.1" <i>GetRegistrationTrafficShaping</i> "2.0.1" <i>GetRemoteSystemDump</i> "1.0.1" <i>GetRFP</i> "3.0.0" <i>GetSecureOMMCertificateServerImport</i> "1.0.3" <i>GetSecurePROVCertificateServerImport</i> "1.0.3" <i>GetSecureSIPCertificateServerImport</i> "2.0.3" <i>GetSoftwareImageURL</i> "1.0.1" <i>GetSpecialBranding</i> "1.0.1" <i>GetUserDataServer</i> "2.0.1" <i>GetWLANProfile</i> "2.0.3" <i>GetXMLApplication</i> "3.0.1" <i>ManualDBBackup</i> "2.0.1" <i>ManualDBRestore</i> "2.0.1" <i>Open</i> "1.0.8" <i>PutFile</i> "1.1.2" <i>SendMessage</i> "1.0.6" <i>SetAlarmTrigger</i> "1.1.4" <i>SetAutoDBBackup</i> "2.0.1" <i>SetCoreDumpURL</i> "1.0.1" <i>SetIMA</i> "2.0.2" <i>SetOMMCertificate</i> "1.0.1" <i>SetOMPURL</i> "1.0.1" <i>SetPBXEnrolment</i> "2.0.1" <i>SetPPFirmwareURL</i> "1.0.1" <i>SetPpProfile</i> "1.0.1" <i>SetPPUser</i> "1.1.4" <i>SetRegistrationTrafficShaping</i> "2.0.1" <i>SetRemoteSystemDump</i> "1.0.1" <i>SetRFP</i> "2.0.0" <i>SetSecureOMMCertificateServerImport</i> "1.0.3" <i>SetSecurePROVCertificateServerImport</i> "1.0.3" <i>SetSecureSIPCertificateServerImport</i> "2.0.3" <i>SetSoftwareImageURL</i> "1.0.1" <i>SetSpecialBranding</i> "1.0.1" <i>SetUserDataServer</i> "2.0.1" <i>SetWLANProfile</i> "2.0.4" <i>SetXMLApplication</i> "3.0.1" </p>	
6.1	<p> <i>EventSARICnf</i> „1.0.0" <i>EventDECTPowerLimitCnf</i> „1.0.0" <i>EventUserDeviceSyncOMMCnf</i> „1.0.0" <i>GetDECTPowerLimit</i> „1.0.0" <i>GetSARI</i> „1.0.0" </p>	New:

	<p> <i>GetUserDeviceSyncOMM „1.0.0"</i> <i>SetDECTPowerLimit „1.0.0"</i> <i>SetSARJ „1.0.0"</i> <i>SetUserDeviceSyncOMM „1.0.0"</i> </p> <p> <i>CreateAccount „1.0.4"</i> <i>CreateAlarmTrigger "1.1.7"</i> <i>CreateLDAP „1.0.2"</i> <i>CreateFixedPP „1.1.7"</i> <i>CreatePPDev "1.0.4"</i> <i>CreatePPUser "1.1.8"</i> <i>CreateXMLApplication "3.0.2"</i> <i>EventAccountCnf „1.0.4"</i> <i>EventAdvancedSIPCnf „3.0.1"</i> <i>EventAlarmTrigger "1.1.6"</i> <i>EventConferenceServerSIPCnf "1.0.1"</i> <i>EventDECTRegDomainCnf "1.0.1"</i> <i>EventFACCnf „1.0.2"</i> <i>EventHealthState"1.1.7"</i> <i>EventLDAPCnf „1.0.1"</i> <i>EventPermissionChange "1.0.3"</i> <i>EventPPDevCnf "1.0.4"</i> <i>EventPpProfileCnf „1.0.1"</i> <i>EventPPUserCnf "1.1.4"</i> <i>EventRFP Cnf "3.0.1"</i> <i>EventXMLApplicationCnf "3.0.2"</i> <i>GetAccount „1.0.4"</i> <i>GetAdvancedSIP „3.0.1"</i> <i>GetAlarmTrigger "1.1.4"</i> <i>GetConferenceServerSIP "1.0.1"</i> <i>GetDECTRegDomain "1.0.1"</i> <i>GetFACList „1.0.2"</i> <i>GetHealthState"1.1.7"</i> <i>GetLDAP „1.0.1"</i> <i>GetPPDev "1.0.4"</i> <i>GetPpProfile „1.0.1"</i> <i>GetPPUser "1.2.6"</i> <i>GetRFPipQuality „1.0.1"</i> <i>GetRFPMediaStreamQuality „1.0.1"</i> <i>GetRFPsyncQuality „1.0.1"</i> <i>GetRFPStatistic „1.0.1"</i> <i>GetSysStatisticMinMax „1.0.1"</i> <i>GetSysStatisticMinMaxRecord „1.0.1"</i> <i>GetSysStatisticMinMaxSummary „1.0.1"</i> <i>GetSysStatisticOccurrence „1.0.1"</i> <i>GetXMLApplication "3.0.2"</i> <i>Open "1.0.12"</i> <i>ResetRFPStatistic „1.0.1"</i> <i>ResetRFPMediaStreamQuality „1.0.1"</i> <i>ResetSysStatistic „1.0.1"</i> <i>SetAccount „1.0.5"</i> </p>	<p>Changed/Bugfix:</p>
--	--	------------------------

<i>SetAdvancedSIP „3.0.1“</i> <i>SetAlarmTrigger "1.1.5"</i> <i>SetConferenceServerSIP "1.0.1"</i> <i>SetDECTRegDomain "1.1.1"</i> <i>SetFAC „1.0.2"</i> <i>SetFACList „1.0.2"</i> <i>SetLDAP „1.0.2"</i> <i>SetPPDev "1.0.4"</i> <i>SetPpProfile „1.0.2"</i> <i>SetPPUser "1.1.7"</i> <i>SetWLANProfile "2.0.5"</i> <i>Subscribe „3.1.1"</i> <i>SetXMLApplication "3.0.2"</i>	
--	--

5.4 OpenMobility® SIP

Following attributes of a DECT phone are used by OpenMobility® SIP (“OpenResp”: “haveSIP”).

Name	Used	CreatePP	SetPP	GetPP
ppn	yes	optional	key	key
name	yes	optional	optional	returned
num	yes	yes	optional	returned
addId	yes	optional	optional	returned
ipei	yes	optional	optional	returned
ac	yes	optional	optional	returned
s	yes	optional	optional	returned
uak	yes	optional	optional	returned
sipAuthId	yes	optional	optional	returned
sipPw	yes	optional	optional	returned
mwiOn	no	n/a	n/a	n/a
mwiT	no	n/a	n/a	n/a

Following Attributes of an RFP are used by OpenMobility® SIP:

Name	Used	CreateRFP	SetRFP	GetRFP
id	yes	optional	key	key
ethAddr	yes	yes	optional	returned
dectOn	yes	optional	optional	returned

wlanOn	yes	optional	optional	returned
licenseRfp	no	n/a	n/a	n/a
location	yes	optional	optional	returned
rpn	yes	ignored	ignored	returned
pagingArea	yes	optional	optional	returned
cluster	yes	optional	optional	returned
x	yes	optional	optional	returned
y	yes	optional	optional	returned
wlanProfile	yes	optional	optional	returned
wlanAntenna	yes	optional	optional	returned
wlanChannel	yes	optional	optional	returned
wlanPower	yes	optional	optional	returned
connected	yes	ignored	ignored	returned
ipAddr	yes	ignored	ignored	returned
dectRunning	yes	ignored	ignored	returned
wlanRunning	yes	ignored	ignored	returned
ommRunning	yes	ignored	ignored	returned

5.5 OpenMobility® License Conditions

The following features/qualities of the OMM releases are licensed:

#	Feature	License Description	License Name	Dependency
1	Installation / System	Enables telephony for a system of a certain size (number of RFPs); linked to a PARK a) PARK for systems up to 256 RFPs b) PARK for systems with more than 256 RFPs	OM System License XXX	
2	Receive Messages / Alarms	Enables receiving	OM Messaging & Alerting System License	1; if messages shall be sent between DECT phone:

#	Feature	License Description	License Name	Dependency
4	Locatable user	Enables locating for a number of users	OM Locating License XXX	1
5	Locating application	Enables the locating application	OM Locating Server License	1; 2; 3; 4

small-RFP installations support built-in licenses with the following characteristics.

#	License Name	Description	Build In License
1	OM System License XXX	Enables telephony for a system of a certain size (number of RFPs)	5 RFPs
2	OM Messaging & Alerting System License	Enables receiving messages/alarms on DECT phones	Yes, system wide but limited to msg. prios "Info", "Low", "Normal" and "High" (no "Emergency", no "Locating Alert")
5	OM Locating Server License	Enables the locating application	No
6	SW Update	Process of updating the license file to enable the system to run with a later major or minor SW release	Free SW update for < 3 RFPs, because activation is not required >= 3 RFPs a new activation is required, because of mandatory activation

5.6 Examples

5.6.1 OMP System Start Up

Request :<Open protocolVersion="44" username="omm" password="*****" OMPClient="1" />

Response:<OpenResp ommStbState="None" ommVersion="OpenMobility Manager SIP-DECT 6.1RC1 Build 1 (DIAG) private patch from ckrohne" ommAxisSpecVersion="6.1" protocolVersion="44" axiClients="6" minPPSwVersion1="5.00.SP5" minPPSwVersion2="6.0.SP1" EULAConfirm="1" uptime="21" haveEnrolmentRFP="1" ommPlatform="linux-pc" haveOmmLogForward="1" haveLicensing="1" haveLocating="1" haveLicensingCodec="1" haveAdditionalSettings="1" haveAutoDB="1" haveLDAP="1" haveDECTRegDomain="1" haveDigitTreatment="1" haveExternalUserData="1" haveFACs="1" haveEnrolmentPP="1" haveIMA="1" haveUMO="1" havePagingAreas="1" haveSIP="1" haveSNMP="1" haveSOSNumberPP="1" haveUnboundDevices="1" haveVideo="1" haveVLAN="1" haveVoiceboxNumber="1" haveWLAN="1" haveXML="1" haveXMLDynamic="1" haveXMLCorpDir="1" haveProtFTP="1" haveProtFTPS="1" haveProtHTTP="1" haveProtHTTPS="1" haveProtSFTP="1" haveProtTFTP="1" havePPFirmwareUpdate="1" haveBluetooth="1" haveConference="1" haveSemiAttendedTransfer="1" haveIPPParamCnf="1" haveSRTP="1" haveHCM="1" haveSecureSIP="1" haveDECTPpSettings="1" haveSpecialBranding="1" havePARKFile="1" haveUserDeviceSync="1" haveSARI="1">

```

<permission>
AllCnfRead</permission>
<permission>
AllCnfWrite</permission>
<permission>
Messaging</permission>
<permission>

```

```

InfoMessaging</permission>
<permission>
Alerting</permission>
<permission>
LocatingAlert</permission>
<permission>
Locating</permission>
<permission>
Monitoring</permission>
<publicKey modulus="CD3A202D953CBF9AD3E7C7E95EC3EBBB59C80C399BC72970D2E4F78ECD0F3528517EFAC47BFD91C4
037A8BC8FE24910C85885072402A87BDC5E70BE9E95534F1" exponent="00010001" />
<accountState id="0" username="user" state="changed" />
<accountState id="1" username="omm" state="changed" />
<accountState id="2" username="root" state="changed" />
</OpenResp>
Request :<Subscribe ><e cmd="On" eventType="SystemState" /></Subscribe>
Response:<SubscribeResp />
Request :<GetVersions />
Response:<GetVersionsResp ActivateRemoteSystemDump="1.0.0" AddUserToConference="2.0.1" CancelMessage="1.0.1"
ChangeUserInConference="2.0.0" ConferenceConfirmation="1.0.0" CreateAccount="1.0.4" CreateACLEntry="
1.0.2" CreateAlarmTrigger="1.0.7" CreateBluetoothBeacon="2.0.0" CreateConference="1.0.0" CreateConfe
renceRoom="1.0.1" CreateDigitTreatment="1.0.0" CreateFixedPP="1.1.7" CreateLDAP="1.0.2" CreatePPDev=
"1.0.4" CreatePPUser="1.1.8" CreateRFP="2.0.0" CreateSite="1.0.1" CreateVideoDev="2.1.0" CreateWLANP
rofile="2.0.3" CreateXMLApplication="3.0.2" DeleteAccount="1.0.0" DeleteACLEntry="1.0.2" DeleteAlarm
Trigger="1.0.2" DeleteBluetoothBeacon="1.0.0" DeleteConference="1.0.0" DeleteConferenceRoom="1.0.0"
DeleteDigitTreatment="1.0.0" DeleteEventLogBuffer="1.0.0" DeleteLDAP="1.0.0" DeleteMessage="1.0.1" D
eletePPDev="1.0.0" DeletePPUser="1.0.1" DeleteRFPCaptureListElem="1.0.1" DeleteRFPCaptureList="1.0.1
" DeleteRFP="1.0.1" DeleteSite="1.0.0" DeleteUserFromConference="1.0.0" DeleteVideoDev="1.0.0" Delet
eWLANProfile="1.0.0" DeleteXMLApplication="1.0.0" EventAccountCnf="1.0.4" EventAccountSummary="1.0.0
" EventACLcnf="1.0.2" EventAdditionalSettingsCnf="1.0.0" EventAddUserToConference="2.0.1" EventAdvan
cedSIPcnf="3.0.1" EventAlarmCallProgress="1.0.0" EventAlarmTriggerCnf="1.0.0" EventAlarmTrigger="1.0
.6" EventAutoDBBackupCnf="2.0.1" EventAutoDBBackupFileNameCnf="1.0.0" EventBackupSIPcnf="1.0.0" Even
tBasicSIPcnf="3.0.0" EventBluetoothBeaconCnf="2.0.0" EventBluetoothBeaconSummary="1.0.0" EventBlueto
othClientStatistic="1.0.0" EventBluetoothGlobalSettingsCnf="1.0.0" EventBluetoothSensitivityCnf="1.0
.0" EventChangeUserInConference="2.0.0" EventConferenceRelease="1.0.0" EventConferenceRequest="1.0.0
" EventConferenceRoomCnf="1.0.1" EventConferenceServerSIPcnf="1.0.1" EventConfigURLcnf="1.0.0" Event
CoreDumpURLcnf="1.0.1" EventCreateConference="1.0.0" EventDbTransferState="1.0.2" EventDECTAuthCodeC
nf="1.0.0" EventDECTEncryptionCnf="1.0.0" EventDECTPagingAreaSizeCnf="1.0.0" EventDECTPpSettingsCnf=
"2.0.0" EventDECTRegDomainCnf="1.0.1" EventDECTSubscriptionMode="1.0.0" EventDeleteConference="1.0.0
" EventDeleteUserFromConference="1.0.0" EventDevAutoCreateCnf="1.0.0" EventDigitTreatmentCnf="1.0.0"
EventDTMFcnf="1.0.0" EventEULAConfirmCnf="1.0.0" EventEventLogEntry="1.0.1" EventFACcnf="1.0.2" Eve
ntFACPrefixCnf="1.0.0" EventFreeConferenceChannels="1.0.0" EventHealthState="1.1.7" EventIMACnf="2.0
.1" EventIntercomCallHandlingSIPcnf="1.0.0" EventLDAPcnf="1.0.1" EventLicenseCnf="2.0.1" EventLicens
edCodeLines="1.0.0" EventLicenseFile="1.0.0" EventLicenseServerListCnf="1.0.0" EventMessageConfirma
tion="1.0.0" EventMessageProgress="1.0.0" EventMessageQueueEmpty="1.0.0" EventMessageSend="1.0.5" Ev
entNetParamsCnf="1.0.0" EventOMMCertificateCnf="1.0.1" EventPARKcnf="1.0.0" EventPARKFromServerResul
t="1.0.0" EventPermissionChange="1.0.3" EventPortRangeSIPcnf="1.0.0" EventPositionHistory="1.0.0" Ev
entPositionInfo="1.0.0" EventPositionRequest="1.0.0" EventPositionTrack="1.0.0" EventDECTPowerLimitC
nf="1.0.0" EventPPcnf="1.0.0" EventPPDevCnf="1.0.4" EventPPDevSummary="1.0.0" EventPPFirmwareUpdateC
nf="1.0.1" EventPPFirmwareUpdateOverview="1.0.0" EventPPFirmwareUpdateStatus="1.0.1" EventPPFirmware
URLcnf="1.0.1" EventPPLLoginVariantCnf="1.0.0" EventPpProfileCnf="1.0.1" EventPPState="1.1.2" EventPP
Transaction="1.0.0" EventPPUserCnf="1.1.7" EventPPUserSummary="1.0.2" EventPreserveUserDeviceRelatio
nCnf="1.0.0" EventReadyForConferencing="1.0.0" EventRegistrationTrafficShapingCnf="2.0.1" EventRemot
eAccessCnf="1.0.0" EventRemoteSystemDumpCnf="1.0.1" EventRestrictedSubscriptionDurationCnf="1.0.0" E
ventRFPCaptureCnf="1.1.0" EventRFPCapture="1.0.2" EventRFPcnf="3.0.1" EventRFPConnectAttempt="1.1.1"
EventRFPDetails="1.1.3" EventRFPipQuality="1.0.0" EventRFPMediaStreamQuality="1.0.0" EventRFPState=
"1.1.3" EventRFPSummary="1.0.3" EventRFPsyncQuality="1.0.0" EventRFPsyncRel="1.0.0" EventRTPcnf="3.0
.0" EventRTPConferenceStreamChg="1.0.0" EventSARICnf="1.0.0" EventSecureSIPcnf="2.0.0" EventSecureOM
MCertificateServerImportCnf="1.0.3" EventSecurePROVCertificateServerImportCnf="1.0.3" EventSecureSIP
CertificateCnf="1.1.0" EventSecureSIPCertificateServerImportCnf="2.0.3" EventSiteCnf="1.0.1" EventSi
teSummary="1.0.1" EventSNMPCnf="1.0.0" EventSoftwareUpdateCnf="1.0.1" EventSpecialBrandingCnf="1.0.1
" EventStbStateChange="1.0.0" EventSuplServCnf="1.0.5" EventSyslogServerCnf="1.0.1" EventSystemCred

```

```

ntialsCnf="1.0.0" EventSystemNameCnf="1.0.0" EventSystemProvUpdTrigCnf="1.0.0" EventSysToneSchemeCnf
="1.0.0" EventSysVoiceboxNumCnf="1.0.0" EventUsedConfigURL="1.0.0" EventUserDataImport="1.0.0" Event
UserDataServerCnf="2.0.1" EventUserDeviceSyncOMMCnf="1.0.0" EventUserMonitoringCnf="1.0.0" EventVide
oDevCnf="2.1.0" EventVideoDevLink="1.0.0" EventVideoDevSummary="1.0.0" EventWLANClient="1.0.1" Event
WLANProfileCnf="2.0.2" EventWLANRegDomainCnf="1.0.0" EventXMLApplicationCnf="3.0.2" GenerateHealthSt
ateAlarmTriggers="1.0.0" GetAccount="1.0.4" GetAccountSummary="1.0.1" GetACLEntry="1.0.1" GetActiveP
PDev="1.0.0" GetAdditionalSettings="1.0.0" GetAdvancedSIP="3.0.1" GetAlarmTrigger="1.1.4" GetAlarmTr
iggerSummary="1.0.0" GetAutoDBBackupFileName="1.0.0" GetAutoDBBackup="2.0.1" GetBackupSIP="1.0.0" Ge
tBasicSIP="3.0.0" GetBluetoothBeacon="2.0.0" GetBluetoothBeaconSummary="1.0.0" GetBluetoothClientSta
tistic="1.0.0" GetBluetoothGlobalSettings="1.0.0" GetBluetoothSensitivity="1.0.0" GetConferenceRoom=
"1.0.1" GetConferenceServerSIP="1.0.1" GetConfigURL="1.0.0" GetCoreDumpURL="1.0.1" GetDbTransferStat
e="1.0.1" GetDECTAuthCode="1.0.0" GetDECTEncryption="1.0.0" GetDECTPagingAreaSize="1.0.0" GetDECTPps
ettings="2.0.0" GetDECTRegDomain="1.0.1" GetDECTSubscriptionMode="1.0.0" GetDevAutoCreate="1.0.0" Ge
tDigitTreatment="1.0.0" GetDigitTreatmentSummary="1.0.0" GetDTMP="1.0.0" GetEULAConfirm="1.0.0" GetE
ventLogBuffer="1.0.1" GetFACList="1.0.2" GetFACPrefix="1.0.0" GetFile="1.0.1" GetFlashMemUsage="1.0.
0" GetFreeConferenceChannels="1.0.0" GetG729ChannelsForConference="1.0.0" GetHealthState="1.1.7" Get
IMA="2.0.1" GetIntercomCallHandlingSIP="1.0.0" GetLastPPDevAction="1.0.0" GetLDAP="1.0.1" GetLicen
seCodecLines="1.0.0" GetLicense="2.0.1" GetLicenseServerList="1.0.0" GetNetParams="1.0.0" GetOMMCerti
ficate="1.0.1" GetPARK="1.0.0" GetPortRangeSIP="1.0.0" GetPPDev="1.0.4" GetPPDevSummary="1.0.1" GetP
PPFirmwareUpdate="1.0.1" GetPPFirmwareUpdateOverview="1.0.1" GetPPFirmwareUpdateStatus="1.0.1" GetPP
FirmwareURL="1.0.1" GetPPLoginVariant="1.0.0" GetPpProfile="1.0.1" GetPPPState="1.1.1" GetDECTPowerLim
it="1.0.0" GetPPUser="1.2.6" GetPPUserSummary="1.0.2" GetPreserveUserDeviceRelation="1.0.0" GetPubli
cKey="1.0.0" GetReadyForConferencing="1.0.0" GetRegistrationTrafficShaping="2.0.1" GetRemoteAccess="
1.0.0" GetRemoteSystemDump="1.0.1" GetRestrictedSubscriptionDuration="1.0.0" GetRFPCapture="1.1.0" G
etRFPCaptureList="1.1.3" GetRFP="3.0.0" GetRFPQuality="1.0.1" GetRFPMediaStreamQuality="1.0.1" Get
RFPStatisticConfig="1.0.0" GetRFPStatistic="1.0.1" GetRFPSummary="1.0.3" GetRFPsSync="1.0.0" GetRFP
sSyncQuality="1.0.1" GetRTP="3.0.0" GetSARI="1.0.0" GetSecureSIP="2.0.0" GetSecureOMMCertificateServerI
mport="1.0.3" GetSecurePROVCertificateServerImport="1.0.3" GetSecureSIPCertificate="1.1.0" GetSecure
SIPCertificateServerImport="2.0.3" GetSite="1.0.1" GetSiteSummary="1.0.2" GetSNMP="1.0.0" GetSoftwar
eUpdate="1.0.0" GetSpecialBranding="1.0.1" GetStbState="1.0.0" GetSuplServ="1.0.5" GetSyslogServer="
1.0.1" GetSysStatisticConfig="1.0.0" GetSysStatisticMinMax="1.0.1" GetSysStatisticMinMaxRecord="1.0.
1" GetSysStatisticMinMaxSummary="1.0.1" GetSysStatisticOccurrence="1.0.1" GetSystemCredentials="1.0.
0" GetSystemName="1.0.0" GetSystemProvUpdTrig="1.0.0" GetSysToneScheme="1.0.0" GetSysVoiceboxNum="1.
0.0" GetUsedConfigURL="1.0.0" GetUserDataServer="2.0.1" GetUserDeviceSyncOMM="1.0.0" GetUserMonitori
ng="1.0.0" GetVersions="2.1.1" GetVideoDev="2.1.0" GetVideoDevLink="1.0.0" GetVideoDevSummary="1.0.0
" GetWLANClients="1.0.0" GetWLANProfile="2.0.3" GetWLANRegDomain="1.0.0" GetWLANRegDomainList="1.0.0
" GetXMLApplication="3.0.2" Limits="2.0.2" ManualDBBackup="2.0.1" ManualDBRestore="2.0.1" ManualUser
DataImport="1.0.0" Open="1.0.12" PARKFromServer="1.0.0" Ping="1.0.1" PutFile="1.1.2" RequestPositionI
nfo="1.0.0" ResetRFPMediaStreamQuality="1.0.1" ResetRFPStatistic="1.0.1" ResetSysStatistic="1.0.1" S
endMessage="1.0.6" SetAccount="1.0.5" SetACLEntry="1.0.0" SetAdditionalSettings="1.0.0" SetAdvancedS
IP="3.0.1" SetAlarmTrigger="1.0.5" SetAutoDBBackup="2.0.1" SetBackupSIP="1.0.0" SetBasicSIP="2.0.0"
SetBluetoothBeacon="2.0.0" SetBluetoothGlobalSettings="1.0.0" SetBluetoothSensitivity="1.0.0" SetCon
ferenceRoom="1.0.1" SetConferenceServerSIP="1.0.1" SetConfigURL="1.0.0" SetCoreDumpURL="1.0.1" SetDE
CTAuthCode="1.0.0" SetDECTEncryption="1.0.1" SetDECTPagingAreaSize="1.0.0" SetDECTPpsSettings="2.0.0"
SetDECTRegDomain="1.1.1" SetDECTSubscriptionMode="1.0.1" SetDevAutoCreate="1.0.0" SetDigitTreatment
="1.0.0" SetDTMP="1.0.0" SetEULAConfirm="1.0.0" SetFAC="1.0.2" SetFACList="1.0.2" SetFACPrefix="1.0.
0" SetIMA="2.0.2" SetIntercomCallHandlingSIP="1.0.0" SetLDAP="1.0.2" SetLicenseServerList="1.0.0" Se
tNetParams="1.0.0" SetOMMCertificate="1.0.1" SetPARK="1.0.0" SetPortRangeSIP="1.0.0" SetPP="1.0.0" S
etPPDev="1.0.4" SetPPFirmwareUpdate="1.0.0" SetPPFirmwareURL="1.0.1" SetPpLink="1.0.0" SetPPLoginVar
iant="1.0.1" SetPpProfile="1.0.2" SetPpUserDevRelation="1.0.0" SetPPUser="1.1.7" SetDECTPowerLimit="
1.0.0" SetPPUserTracking="1.0.0" SetPreserveUserDeviceRelation="1.0.0" SetRegistrationTrafficShaping
="2.0.1" SetRemoteAccess="1.0.0" SetRemoteSystemDump="1.0.1" SetRestrictedSubscriptionDuration="1.0.
0" SetRFPCapture="1.1.0" SetRFP="2.0.0" SetRTPConferenceStreamChg="1.0.0" SetRTP="3.0.0" SetSARI="1.
0.0" SetSecureSIP="2.0.0" SetSecureOMMCertificateServerImport="1.0.3" SetSecurePROVCertificateServer
Import="1.0.3" SetSecureSIPCertificate="1.1.0" SetSecureSIPCertificateServerImport="2.0.3" SetSite="
1.0.1" SetSNMP="1.0.0" SetSpecialBranding="1.0.1" SetSuplServ="1.0.3" SetSyslogServer="1.0.1" SetSys
temCredentials="1.0.0" SetSystemName="1.0.0" SetSystemProvUpdTrig="1.0.0" SetSysToneScheme="1.0.0" S
etSysVoiceboxNum="1.0.0" SetUserDataServer="2.0.1" SetUserDeviceSyncOMM="1.0.0" SetUserMonitoring="1
.0.0" SetVideoDev="2.1.0" SetWLANProfile="2.0.5" SetWLANRegDomain="1.1.0" SetXMLApplication="3.0.2"
SoftwareUpdate="1.0.2" Subscribe="3.1.1" SystemRestart="1.0.1" />
Request :<Limits />
Response:<LimitsResp axiClients="100" codec="5" ldapServer="5" licLatency="43200" ppnNum="10000" records="20"

```

```

    rfpNum="4096" site="250" dnsServer="3" ntpServer="3" ssidWlan="4" ssidWlanKey="4" trigger="100" url
    Len="513" userId="100" wlanMacFilter="512" xmlBuiltInAppl="7" xmlDynAppl="10" xmlCorpDirAppl="5" wla
    nClients="262144" wlanProfiles="20" bluetoothBeacons="28672" bluetoothNeighbours="8" bluetoothRssiVa
    lues="6" conferenceRooms="100" ppProfileNum="20" limitRTT1="25" limitRTT2="50" limitRTT3="150" limit
    RTT4="500" certificateNum="10" ppVideoDevNum="10" />
Request :<Subscribe ><e cmd="On" eventType="PPDevState" ppn="-1" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="PPSipState" ppn="-1" /></Subscribe>
Response:<SubscribeResp />
Request :<GetSystemName />
Response:<GetSystemNameResp name="OMM" />
Request :<Subscribe ><e cmd="On" eventType="SystemCnf" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="AutoDBCnf" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="DECTSubscriptionMode" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="FACCnf" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="PPFirmwareUpdateCnf" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="PPFirmwareUpdateOverview" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="ParkServerResult" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="Roaming" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="SiteCnf" /></Subscribe>
Response:<SubscribeResp />
Request :<GetSite id="0" maxRecords="20" />
Response:<GetSiteResp>
    <site id="1" name="default" nRFPs="5" wideband="0" srtp="Disabled" dectSecurity="0" videoStreaming="
    1" />
    <site id="2" name="Kiel" nRFPs="0" wideband="0" srtp="Disabled" dectSecurity="0" videoStreaming="0"
    />
</GetSiteResp>
Request :<Subscribe ><e cmd="On" eventType="AccountCnf" /></Subscribe>
Response:<SubscribeResp />
Request :<GetAccount id="0" maxRecords="20" />
Response:<GetAccountResp>
    <account id="0" username="user" password="KPCIXAW1L3ZLRkw6R6nonYdL5VUDeLkH28Vp5krbnpn2XxYHHvawIUOX7s
    qTetokEoxT/lFnxJeszP2S7e5Lsg==" expire="0" active="1" aging="none" state="changed">
    <permission>
    AllCnfRead</permission>
    </account>
    <account id="1" username="omm" password="u2kCqZBmYJj1w12GIB2h1/h1YU5KeAIws22ucIGA9PkXVJC8Cm/7XRCV7C
    0gnPglZnjjaBfXcs88/4VprzHmQ==" expire="0" active="1" aging="none" state="changed">
    <permission>
    AllCnfRead</permission>
    <permission>
    AllCnfWrite</permission>
    <permission>
    Messaging</permission>
    <permission>
    InfoMessaging</permission>
    <permission>
    Alerting</permission>
    <permission>
    LocatingAlert</permission>
    <permission>
    Locating</permission>
    <permission>

```

```

Monitoring</permission>
</account>
<account id="2" username="root" password="wWtlM8q846KXkHeU+3tdM1ZRRIXEJPNu1beI7FFj5Ysv00MhxZrCAZt78
rXodX25VCRmmyVb7MTM3nckhiuw==" expire="0" active="1" aging="none" state="changed">
<permission>
AllCnfRead</permission>
<permission>
AllCnfWrite</permission>
</account>
<account id="3" username="ckrohne" password="Mqvo8Ayd/XkPJHUQGu9SblCndnciwJPe3ebZrZhbHX8FpRo0lAlmIX7
BPczblCWrgLv4jJanv01AHW3HWBUGAQ==" expire="0" active="1" aging="none" state="dontCare">
<permission>
AllCnfRead</permission>
<permission>
AllCnfWrite</permission>
<permission>
Messaging</permission>
<permission>
InfoMessaging</permission>
<permission>
Monitoring</permission>
</account>
</GetAccountResp>
Request :<Subscribe ><e cmd="On" eventType="RFPsSummary" /></Subscribe>
Response:<SubscribeResp />
Request :<GetRFPsSummary />
Response:<GetRFPsSummaryResp nRFPs="5" idFirst="1" nConnected="2" wrongBrandedRFPs="0" wrongStandbyRFPs="0" wr
ongVersionedRFPs="0" newAvailSWRFPs="0" DecryptedDECTRFPs="0" usbOverloads="0" DECTactivatedRFPs="1"
DECTactiveRFPs="0" advancedFeaturesErrorRFPs="0" usedDECTclusters="1" usedPagingAreas="1" WLANactiv
atedRFPs="1" WLANrunningRFPs="1" usedWLANprofiles="1" />
Request :<Subscribe ><e cmd="On" eventType="PPUserSummary" /></Subscribe>
Response:<SubscribeResp />
Request :<GetPPUserSummary />
Response:<GetPPUserSummaryResp nRecords="3" uidFirst="1" nLocatable="0" nBtLocatable="0" nMsgSend="2" nSipReg
istration="0" usersActiveMonitored="0" usersPassiveMonitored="0" usersWarned="0" usersUnavailable="0
" usersEscalated="0" />
Request :<Subscribe ><e cmd="On" eventType="PPDevSummary" /></Subscribe>
Response:<SubscribeResp />
Request :<GetPPDevSummary />
Response:<GetPPDevSummaryResp nRecords="3" ppnFirst="116" subscribedDevs="2" />
Request :<Subscribe ><e cmd="On" eventType="SiteSummary" /></Subscribe>
Response:<SubscribeResp />
Request :<GetSiteSummary />
Response:<GetSiteSummaryResp nRecords="2" idFirst="1" rfpSiteNum="1" g722SiteNum="0" dectSecuritySiteNum="0"
srtpSiteNum="0" videoStreamingSiteNum="1" />
Request :<Subscribe ><e cmd="On" eventType="VideoDevSumCnf" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="VideoDevSumState" /></Subscribe>
Response:<SubscribeResp />
Request :<GetVideoDevSummary />
Response:<GetVideoDevSummaryResp nRecords="0" idFirst="4294967295" enabledNum="0" stateUnpluggedNum="0" state
InactiveNum="0" stateActiveNum="0" stateFailedNum="0" />
Request :<Subscribe ><e cmd="On" eventType="ConferenceCnf" /></Subscribe>
Response:<SubscribeResp />
Request :<GetConferenceRoom id="0" maxRecords="20" />
Response:<GetConferenceRoomResp>
<room id="0" name="" conferenceId="9999" sipAuthId="" sipPw="cc6LbGuntqlukie/n4lMEa4MqmMKnmRZe9hacbA
PdyfFO8cv6Z51pwA5b/4G/SIRKu5aGD/Tookn6wvtiSiOAA==" fixedSipPort="4711" calculatedSipPort="0" />
</GetConferenceRoomResp>
Request :<Subscribe ><e cmd="On" eventType="Conference" /></Subscribe>
Response:<SubscribeResp />
Request :<GetFreeConferenceChannels />

```

```

Response:<GetFreeConferenceChannelsResp freeChannels="15" totalChannels="15" />
Request :<GetSoftwareUpdate />
Response:<GetSoftwareUpdateResp timedUpdate="0" allAtOnce="0" hour="0" minute="0" />
Request :<Subscribe ><e cmd="On" eventType="WLANProfileCnf" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="ACLCnf" /></Subscribe>
Response:<SubscribeResp />
Request :<GetWLANProfile id="1" maxRecords="20" />
Response:<GetWLANProfileResp>
  <profile id="1" enable="1" beaconIntval="100" dtimIntval="5" rtsThreshold="2346" fragThreshold="2346
  " nRFPs="0" maxRate="54000" interferenceAvoidance="0" mode="B" qos="None" profileType="RFP42">
  <ssid ssid="meins" enable="1" bssIsolation="0" useACL="0" hiddenSSID="0" security="WEP" privacy="0"
  wepKey1="Wn8bw" wepKey2="CX2BO" wepKey3="TwGK7" wepKey4="EKm6A" defaultWepKey="1" wepKeyHex="0" key
  Length="64" />
  </profile>
  <profile id="2" enable="1" beaconIntval="100" dtimIntval="5" rtsThreshold="2346" fragThreshold="2346
  " nRFPs="1" mode="G" qos="WME" profileType="RFP43">
Request :<GetRFPCaptureList />
  <ssid ssid="MyAccessPoint" enable="1" bssIsolation="0" useACL="0" hiddenSSID="0" security="WEP" pri
  vacy="1" wepKey1="ckomm" defaultWepKey="1" wepKeyHex="0" keyLength="64" />
  </profile>
  <profile id="3" enable="0" beaconIntval="100" dtimIntval="5" rtsThreshold="2347" fragThreshold="2346
  " nRFPs="0" maxRate="54000" interferenceAvoidance="0" mode="G" qos="None" profileType="RFP42">
  <ssid ssid="test" enable="0" bssIsolation="0" useACL="0" hiddenSSID="0" security="Open" />
  </profile>
  <profile id="6" enable="1" beaconIntval="100" dtimIntval="5" rtsThreshold="7" fragThreshold="256" nR
  FPs="0" mode="N" qos="WME-DiffServ" profileType="RFP43" />
  </GetWLANProfileResp>
Response:<GetRFPCaptureListResp errCode="ENoEnt" />
Request :<Subscribe ><e cmd="On" eventType="PPDevCnf" ppn="-1" /></Subscribe>
Request :<Subscribe ><e cmd="On" eventType="PPUserCnf" uid="-1" /></Subscribe>
Response:<SubscribeResp />
Request :<GetPPDev ppn="0" maxRecords="20" />
Response:<SubscribeResp />
Response:<GetPPDevResp>
  <pp ppn="116" ppnHex="0x74" relType="Fixed" uid="503" uidHex="0x1f7" timeStamp="1433773006" ipei="00
  100 0000499 8" ac="" s="No" encrypt="0" capMessaging="0" capMessagingForInternalUse="0" capEnhLocati
Request :<Subscribe ><e cmd="On" eventType="AlarmTriggerCnf" /></Subscribe>
  ng="0" capBluetooth="0" ethAddr="" hwType="Unknown" ppProfileCapability="0" ppDefaultProfileLoaded="
  0" subscribeToPARIOnly="0" ommId="00000000" ommIdAck="00000000" autoCreate="0" modicType="AA=" loca
  tionData="AAAAAAA" roaming="RoamingIdle" dectIeFixedId="AAAAAAA" subscriptionId="AAAAAAA"
  />
  <pp ppn="4576" ppnHex="0x11e0" relType="Unbound" uid="0" uidHex="0x0" timeStamp="1434521613" ipei="0
  3586 0677843 8" ac="" s="Yes" uak="3FE502FB01147A004E073B7F20CE1C75" encrypt="1" capMessaging="1" ca
  pMessagingForInternalUse="0" capEnhLocating="1" capBluetooth="1" ethAddr="00:30:42:15:A3:B6" hwType=
  "650c" ppProfileCapability="1" ppDefaultProfileLoaded="1" subscribeToPARIOnly="0" ommId="10187343" o
  mmIdAck="10187343" autoCreate="1" modicType="QQ=" locationData="AvsBFHoA" roaming="RoamingComplete"
  dectIeFixedId="BqCgEBhzQgA=" subscriptionId="BQDgKlftAAA=" />
  <pp ppn="4577" ppnHex="0x11e1" relType="Unbound" uid="0" uidHex="0x0" timeStamp="1434521622" ipei="0
  3586 0677920 4" ac="" s="Yes" uak="A4E502FB01147A00A49624E1A378D72C" encrypt="1" capMessaging="1" ca
  pMessagingForInternalUse="0" capEnhLocating="1" capBluetooth="1" ethAddr="00:30:42:15:A4:03" hwType=
  "650c" ppProfileCapability="1" ppDefaultProfileLoaded="1" subscribeToPARIOnly="0" ommId="10187343" o
  mmIdAck="10187343" autoCreate="1" modicType="QQ=" locationData="AvsBFHoA" roaming="RoamingComplete"
  dectIeFixedId="BqCgEBhzQgA=" subscriptionId="BQDgKlggAAA=" />
  </GetPPDevResp>
Request :<GetAlarmTrigger id="0" maxRecords="20" />
Response:<SubscribeResp />
Response:<GetAlarmTriggerResp>
  <trigger id="0" triggerId="SOS" fac="SOS" comment="" num="" />
  <trigger id="1" triggerId="MANDOWN" fac="MANDOWN" comment="" num="" />
  <trigger id="2" triggerId="2" fac="*" comment="test" num="4711" />
  </GetAlarmTriggerResp>

```

```

Request :<Subscribe ><e cmd="On" eventType="DigitTreatmentCnf" /></Subscribe>
Response:<SubscribeResp />
Request :<GetDigitTreatment id="0" maxRecords="20" />
Request :<GetLDAP />
Response:<GetDigitTreatmentResp>
    <rule id="0" externalPattern="+49" internalPattern="00" direction="Directory" directory="0" />
    <rule id="1" externalPattern="54321" internalPattern="12345" direction="Both" directory="0">
    <sites siteId="1" />
    </rule>
    </GetDigitTreatmentResp>
Response:<GetLDAPResp />
Request :<GetXMLApplication />
Request :<Subscribe ><e cmd="On" eventType="PPHCM" /></Subscribe>
Response:<GetXMLApplicationResp>
    <xmlAppl id="0" type="BuiltIn" enable="0" name="callerList" corpDirOrder="0">
    <url protocol="HTTP" host="10.100.88.95" port="0" path="sipDectTerm.xml?requestType=localFeature&
    ;EuroFea=219&key=20" username="" password="JUGZbmSxdNAbNcosdoWY4U/Ob1pVIVyDWEsjQcg11X8hpXDLcujce
    EKRVEqNL5LaedyZFnd3SGhPU1NUQxd5XQ==" />
    </xmlAppl>
    <xmlAppl id="1" type="BuiltIn" enable="1" name="redialList" corpDirOrder="0">
    <url protocol="HTTP" host="10.100.88.95" port="3197" path="sipDectTerm.xml?requestType=localFeature&
    amp;EuroFea=215" username="" password="FO+mrxcwKDYjctm2bc0NvnDcfPRkAG74ZhJ6Gv/AdnNlfJ2/IeT4UblAZMcPpo
    cLs955oWr3mClQCn2n6uxjxLQ==" />
    </xmlAppl>
    <xmlAppl id="2" type="BuiltIn" enable="1" name="userPresence" corpDirOrder="0">
    <url protocol="HTTP" host="10.100.88.95" port="0" path="sipDectTerm.xml?requestType=localFeature&
    ;EuroFea=372" username="" password="Rz7dOdve9OYrhMjfq1+OApAlcNUEvNdRmCMCTC9xA44xm8Vxj9jj+y58XtZGfTrM
    4/k9F5VrSMU654zZJdalPg==" />
    </xmlAppl>
    <xmlAppl id="3" type="BuiltIn" enable="0" name="systemApplMenu" corpDirOrder="0">
    <url protocol="HTTP" host="10.100.88.95" port="0" path="sipDectTerm.xml?requestType=localFeature&
    ;EuroFea=208" username="" password="P+y6VnkRYby2ELQAahNAHdfFpauPDmwpDMe5bXPRPI/3V1TkPVO5BjvxT9KrUpBM
    1tYgd/YyCSG01IguWPYLQ==" />
    </xmlAppl>
    <xmlAppl id="4" type="BuiltIn" enable="0" name="eventActions" corpDirOrder="0">
    <url protocol="HTTP" host="10.100.88.95" port="0" path="sipDectTerm.xml?startup={boot}&registered=
    {reg}&disconnected={dis}" username="" password="WNQM6KJmBzjd/u2e6Q25jOuK6jJt/Sk0KTufbqCiXiQiNi+
    pnpCUvJoOy+bNzLCR+5o7RYX3HAhX0Sy82Xsb0Pw==" />
    </xmlAppl>
    <xmlAppl id="5" type="BuiltIn" enable="0" name="featureAccessCodes" corpDirOrder="0">
    <url protocol="HTTP" host="10.103.38.11" port="0" path="ppxml/fac.php?na={subsc}&pp={ppn}&fa
    c={fac}" username="" password="Vfrd6u500FNrtAiGZMTmdi0kJzEV9eHu2TIYs18s1XqLjmqWcbQHkbJ06DD3gbFJM1TI
    pfvCMN6jjiVOzEHbg==" />
    </xmlAppl>
    <xmlAppl id="6" type="BuiltIn" enable="0" name="callCompletion" corpDirOrder="0">
    <url protocol="HTTP" host="10.100.88.95" port="0" path="sipDectTerm.xml?requestType=localFeature&
    ;EuroFea=390" username="" password="D/NtEe+JjE2og3ytOxWs+y3jSSP1N0SZpg0zVaQNZiTQMT3cF0d14RL6DEeBricl
    HQDp/xH/B5/5cOWI68BggA==" />
    </xmlAppl>
    <xmlAppl id="7" type="Dynamic" enable="0" name="Julians Appl" corpDirOrder="0">
    <url protocol="HTTPS" host="10.103.38.11" port="0" path="ppxml/index.php" username="user" password="
    IfTfWkHF8hJ8zhf6x+KheFnpmurQBzkj1rA/y3R6t/nvae1TL7upSNYk6TwausrmLSi7ZuBy+LNTY4Is7Lk==" />
    </xmlAppl>
    <xmlAppl id="8" type="Dynamic" enable="0" name="Page wise scrolling" corpDirOrder="0">
    <url protocol="HTTP" host="ber-rd5022" port="0" path="omm.mghc.PageWiseScrolling?na={subsc}" usernam
    e="" password="L8PkE8QSiHqrxfmG0OTNei3c1DQOrnhQc3Fqyw/UnrCrzvjKjzltHVWbkTbBFDR09MrCl4uusCUMNy8BDBrt
    w==" />
    </xmlAppl>
    <xmlAppl id="9" type="Dynamic" enable="0" name="Line wise scrolling" corpDirOrder="0">
    <url protocol="HTTP" host="ber-rd5022" port="0" path="omm.mghc.LineWiseScrolling?na={subsc}" usernam
    e="" password="uDUXt3oQ/JSPh11zZixFKB8QBwjvP4xMpZx67msdyDQSm15dkxVjaINSmKlBsjGFftVr9JQMDRZHcvaFj+Vkl
    g==" />

```

```

</xmlAppI>
<xmlAppI id="10" type="Dynamic" enable="0" name="SIP Login/Logout" corpDirOrder="0">
<url protocol="HTTP" host="10.103.38.11" port="0" path="ppxml/index.php?type=xmllogin" username="" p
assword="nh7l20dvY1T40iCoJ50JcRSu6mJIs+CnlelU0+qiWoh4G6AWkwVSKW7lTQiOlii2WYl8PKztUsZewLtqppL0Q==" /
>
</xmlAppI>
<xmlAppI id="11" type="Dynamic" enable="0" name="CMG DIR (SE)" corpDirOrder="0">
<url protocol="HTTP" host="10.105.21.60" port="0" path="xml/directory/CorpDir.php" username="" passw
ord="Jejs067hRRwI7fpc5AdEWLvG6+AMfEhKuD/o35shHJSh/8+hVIOUXsKm/5Z7sEC3lc2xVgJbz9P9jGqUnq74Qw==" />
</xmlAppI>
<xmlAppI id="12" type="Dynamic" enable="0" name="CorpDir" corpDirOrder="0">
<url protocol="HTTP" host="10.105.21.60" port="0" path="xml/directory/CorpDir.php" username="" passw
ord="Fp+9AnV3WneXnVKtb76iXNacIF/wUjXwPxtL6koNGnH7CeD8rOEsqxQW9dexCEKbgCM4FYwpXMALqx3lPdk4Q==" />
</xmlAppI>
<xmlAppI id="13" type="Dynamic" enable="0" name="OMM Wizard" corpDirOrder="0">
<url protocol="HTTP" host="10.103.38.108" port="0" path="/svn/ommwizard/index.php" username="" passw
ord="mEtu/5LPYQyUeGeQEQGlosAz63nERDeKSlHulJdszLdLTmblYvc1z40IUnlJgxl8+U4fuRvx2InDcIlmMfU2Q==" />
</xmlAppI>
<xmlAppI id="14" type="Dynamic" enable="0" name="Python" corpDirOrder="0">
<url protocol="HTTP" host="172.17.1.109" port="81" path="admin.py" username="" password="ptv66d26hn3
iRemyzio2l4t4Io2UL9Sti/Gv7o5t7+gAKkliV0kt/OZQAc6vPppWwsnlAK37zHKs+w37U0dxg==" />
</xmlAppI>
<xmlAppI id="15" type="Dynamic" enable="0" name="My Application" corpDirOrder="0">
<url protocol="HTTP" host="10.103.35.22" port="0" path="LongItemString" username="" password="TZH++i
BZl+h955EjP29dntBgXSK680VjiEOETX0xQ9FdmrGlVk9ggL56UvRv6tWSxCnalewIMq9OpGyj7HqPXQ==" />
</xmlAppI>
<xmlAppI id="16" type="Dynamic" enable="0" name="SMS" corpDirOrder="0">
<url protocol="HTTP" host="server" port="0" path="Test" username="" password="kduKiNikUyUc3U6BFL5Aur
k9JNXbQlggscblaMxvMfyvirtCZa90Ck89yLMqz1B4y77LD50YvW3KlFkJLTzPsg==" />
</xmlAppI>
</GetXMLApplicationResp>
Response:<SubscribeResp />
Request :<GetPpProfile id="0" />
Request :<Subscribe ><e cmd="On" eventType="EventLog" /></Subscribe>
Request :<GetEventLogBuffer id="0" maxRecords="20" />
Response:<GetPpProfileResp>
<ppProfile id="0" name="Default" timeStamp="1430207616" ppData="UD_ConfigurationName=default&#10;UD_
DispLang=sv&#10;UD_DispFont=small&#10;UD_DispColor=plain&#10;" />
</GetPpProfileResp>
Request :<GetRFPStatisticConfig />
Response:<SubscribeResp />
Response:<GetEventLogBufferResp eof="1">
<e id="0" time="2015/06/26 08:06:24.203" level="GenInfo" name="STB" count="1" msg="STB : 0 OMMs on c
ommand line / OMM does not run in standby mode" />
<e id="1" time="2015/06/26 08:06:24.203" level="GenInfo" name="CNF" count="1" msg="CNF : SIP-DECT 6.
1RC1 Build 1 (DIAG) private patch from ckrohne start up" />
<e id="2" time="2015/06/26 08:06:24.253" level="Event" name="AXI" count="1" msg="AXI : [174] New con
nection from 10.103.35.22:58112" />
<e id="3" time="2015/06/26 08:06:24.387" level="Event" name="AXI" count="1" msg="AXI : [172] New con
nection from 10.103.35.22:58113" />
<e id="4" time="2015/06/26 08:06:24.387" level="MinorErr" name="AXI" count="1" msg="AXI : Open for u
ser/device synchronization client detected" />
<e id="5" time="2015/06/26 08:06:24.753" level="Event" name="AXI" count="1" msg="AXI : [176] New con
nection from 10.103.35.22:58115" />
<e id="6" time="2015/06/26 08:06:24.771" level="Event" name="AXI" count="1" msg="AXI : [178] New con
nection from 10.103.35.22:58116" />
<e id="7" time="2015/06/26 08:06:29.247" level="Event" name="AXI" count="1" msg="AXI : [183] New con
nection from 10.103.35.22:58117" />
<e id="8" time="2015/06/26 08:06:34.368" level="MinorErr" name="IPL" count="1" msg="IPL : RFP(0004)
reconnected" />
<e id="9" time="2015/06/26 08:06:45.341" level="Event" name="AXI" count="1" msg="AXI : [184] New sec
ure connection from 10.103.35.22:56631" />

```



```

    </GetEventLogBufferResp>
Response: <GetRFPStatisticConfigResp>
    <rftpStatHead numElemPerRec="28" recordSets="3" resolution="week" />
    <rftpStatName elemId="0" group="Voice channels" name="Only 2 voice channels free" />
    <rftpStatName elemId="1" group="Voice channels" name="Only 1 voice channels free" />
    <rftpStatName elemId="2" group="Voice channels" name="Voice channels busy" />
    <rftpStatName elemId="3" group="Voice channels" name="Voice channels busy and call rejected" />
    <rftpStatName elemId="4" group="Air channels" name="Only 2-4 air channels free" />
    <rftpStatName elemId="5" group="Air channels" name="Only 1 air channel free" />
    <rftpStatName elemId="6" group="Air channels" name="Air channels busy" />
    <rftpStatName elemId="7" group="Paging" name="Paging queue overflows" />
    <rftpStatName elemId="8" group="Sync" name="Synchronisation losts" />
    <rftpStatName elemId="9" group="Sync" name="Low relations" />
    <rftpStatName elemId="10" group="Sync" name="Offset jumps" />
    <rftpStatName elemId="11" group="RFP health" name="RFP resets" />
    <rftpStatName elemId="12" group="RFP health" name="RFP connection timeouts" />
    <rftpStatName elemId="13" group="BMC: Connections" name="01 - 03" />
    <rftpStatName elemId="14" group="BMC: Connections" name="04 - 06" />
    <rftpStatName elemId="15" group="BMC: Connections" name="07 - 09" />
    <rftpStatName elemId="16" group="BMC: Connections" name="10 - 12" />
    <rftpStatName elemId="17" group="BMC: DSP chan used" name="01 - 02" />
    <rftpStatName elemId="18" group="BMC: DSP chan used" name="03 - 04" />
    <rftpStatName elemId="19" group="BMC: DSP chan used" name="05 - 06" />
    <rftpStatName elemId="20" group="BMC: DSP chan used" name="07 - 08" />
    <rftpStatName elemId="21" group="BMC: Miscellaneous" name="Lost connections" />
    <rftpStatName elemId="22" group="BMC: Miscellaneous" name="MAC reset" />
    <rftpStatName elemId="23" group="BMC: Miscellaneous" name="Reject dummy" />
    <rftpStatName elemId="24" group="BMC: Miscellaneous" name="Ho timer &gt; 150ms" />
    <rftpStatName elemId="25" group="BMC: Frame error rate" name="Bad frames" />
    <rftpStatName elemId="26" group="BMC: Frame error rate" name="Good frames" />
    <rftpStatName elemId="27" group="BMC: Frame error rate" name="Frame error rate in 1/1000" />
    </GetRFPStatisticConfigResp>
Request : <GetPublicKey />
Response: <GetPublicKeyResp modulus="CD3A202D953CBF9AD3E7C7E95EC3EBBB59C80C399BC72970D2E4F78ECD0F3528517EFAC47
    BF9D1C4037A8BC8FE24910C85885072402A87BDC5E70BE9E95534F1" exponent="00010001" />
Request : <GetACLEntry wlanProfile="1" ethAddr="00:00:00:00:00:00" maxRecords="20" />
Response: <GetACLEntryResp errCode="ENoEnt" />
Request : <GetACLEntry wlanProfile="2" ethAddr="00:00:00:00:00:00" maxRecords="20" />
Response: <GetACLEntryResp errCode="ENoEnt" />
Request : <GetACLEntry wlanProfile="3" ethAddr="00:00:00:00:00:00" maxRecords="20" />
Response: <GetACLEntryResp errCode="ENoEnt" />
Request : <GetACLEntry wlanProfile="6" ethAddr="00:00:00:00:00:00" maxRecords="20" />
Response: <GetACLEntryResp errCode="ENoEnt" />
Request : <GetPPState seq="116" ppn="116" />
Response: <GetPPStateResp ppn="116" ppnHex="0x74" onHook="1" registered="0" silentCharging="0" callState="idle
    " swVersion="" bluetooth="0" regServerType="None" seq="116" />
Request : <GetPPState seq="4576" ppn="4576" />
Response: <GetPPStateResp ppn="4576" ppnHex="0x11e0" onHook="1" registered="0" silentCharging="0" callState="n
    one" swVersion="" bluetooth="0" regServerType="None" seq="4576" />
Request : <GetPPState seq="4577" ppn="4577" />
Response: <GetPPStateResp ppn="4577" ppnHex="0x11e1" onHook="1" registered="0" silentCharging="0" callState="n
    one" swVersion="" bluetooth="0" regServerType="None" seq="4577" />
Request : <GetPPUser uid="0" maxRecords="20" />
Response: <GetPPUserResp>
    <user uid="1" uidHex="0x1" relType="Unbound" ppn="0" ppnHex="0x0" timeStamp="1435050382" name="test"
    num="1" hierarchy1="" hierarchy2="" addId="" sipAuthId="" sipPw="g2FIUurj6Nzb94/tRIX0sMlAX9yY1sY40+t
    Tozp3lmK1GBI8S9LLmArmerY7gUHMCMq396tvqH7QTUwslBgA==" sosNum="" manDownNum="" voiceboxNum="" pin=""
    fjaTnS013acV1s5JxBl072BiPjcVSDLmTOA95MfwaVHS1MS9MynbMUCd9vIFfIFVsSk7/8bY36rEtNEo13MQ==" lang="Engl
    ish" forwardState="Off" forwardDest="" forwardTime="0" holdRingBackTime="3" callWaitingDisabled="0"
    autoAnswer="Global" microphoneMute="Global" warningTone="Global" allowBargeIn="Global" trackingActiv
    e="0" locRight="0" locatable="0" msgRight="1" sendVcardRight="1" recvVcardRight="0" keepLocalPB="0"
    vip="0" sipRegisterCheck="0" external="0" BTlocatable="0" BTsensitivity="high" conferenceServerType=

```

```

"Global" conferenceServerURI="" monitoringMode="Off" HAS="Unknown" HSS="Unknown" HRS="Unknown" HCS="
Unknown" SRS="Unknown" SCS="Unknown" CDS="Unknown" HBS="Unknown" BTS="Unknown" SWS="Unknown" CUS="Un
known" allowVideoStream="0" credentialPw="utugris/y4uGj5toerHLkUE2QwGof/4/mULAfPSCfOTHahgCrX7F8jPqdn
23WPmbnZ7HehBstEQGFxhq2BHdfQ==" fixedSipPort="0" calculatedSipPort="0" configurationDataLoaded="0" p
pProfileId="0" ppData="" ppnOld="0" ppnOldHex="0x0" />
<user uid="2" uidHex="0x2" relType="Unbound" ppn="0" ppnHex="0x0" timeStamp="1435227408" name="User
31201" num="31201" hierarchy1="Company1" hierarchy2="Departement1" addId="" sipAuthId="" sipPw="mSQb
YQkBBZ/t37MDYTCoot7vc0oSpt9xjElCWjeSgPWViEUMn8Saujmc8sD4XohwUxbIXGNNIr5oHE3fa+Kig==" sosNum="100" m
anDownNum="112" voiceboxNum="1111" pin="JR92/lwjXSMcI8RhImfcZESuFHUTcSTdn1C6Sd6e3109zMuIaizdJnCte931
+NcHmdkYHr3p700SkQUkb0Sz7g==" lang="German" forwardState="Off" forwardDest="" forwardTime="0" holdRi
ngBackTime="3" callWaitingDisabled="0" autoAnswer="Off" microphoneMute="Off" warningTone="Off" allow
BargeIn="Off" trackingActive="0" locRight="0" locatable="0" msgRight="1" sendVcardRight="1" recvVcar
dRight="0" keepLocalPB="1" vip="0" sipRegisterCheck="0" external="1" BTlocatable="0" BTsensitivity="
medium" conferenceServerType="Global" conferenceServerURI="" monitoringMode="Off" HAS="Unknown" HSS="
Unknown" HRS="Unknown" HCS="Unknown" SRS="Unknown" SCS="Unknown" CDS="Unknown" HBS="Unknown" BTS="U
nknown" SWS="Unknown" CUS="Unknown" allowVideoStream="0" credentialPw="IN+FvoHYq2xGwB6oXLe2C4rdSeWy0
QeyUVa0mW5MRGx3q30oTt93/PRW2q4uPpu7nubtMreUYWIn0lHewmsg==" fixedSipPort="0" calculatedSipPort="506
0" configurationDataLoaded="0" ppProfileId="0" ppData="UD_ConfigurationName=user31201&#10;UD_Displan
g=de&#10;UD_DispFont=small&#10;UD_DispColor=black&#10;" ppnOld="4577" ppnOldHex="0x11e1" />
<user uid="503" uidHex="0x1f7" relType="Fixed" ppn="116" ppnHex="0x74" timeStamp="1433773288" name="
simu_pp_499" num="7499" hierarchy1="" hierarchy2="" addId="" sipAuthId="" sipPw="P6aRuFHVHC9TtXQAeRU
ZDJXC8xfXsWn/ECJ6tdhIMTuQjBQ7d0bRLsZhjacmD/2wa3q6N0v0lme5rs82W5B6kg==" sosNum="" manDownNum="" voice
boxNum="" pin="MDBF7KZUzi7Y7ChJx8Tf6A5vCeOkClSr8rVou7wpvvdnb4jfz0LLeC7opzkyXYqC6OuPKZKVAONZ6wBnpgmm5
Q==" lang="English" forwardState="Off" forwardDest="" forwardTime="0" holdRingBackTime="3" callWaiti
ngDisabled="0" autoAnswer="Global" microphoneMute="Global" warningTone="Global" allowBargeIn="Global
" trackingActive="0" locRight="0" locatable="0" msgRight="0" sendVcardRight="0" recvVcardRight="0" k
eepLocalPB="0" vip="0" sipRegisterCheck="0" external="0" BTlocatable="0" BTsensitivity="high" confer
enceServerType="Global" conferenceServerURI="" monitoringMode="Off" HAS="Unknown" HSS="Unknown" HRS=
"Unknown" HCS="Unknown" SRS="Unknown" SCS="Unknown" CDS="Unknown" HBS="Unknown" BTS="Unknown" SWS="U
nknown" CUS="Unknown" allowVideoStream="0" credentialPw="s5FE2qomZTA7HKyOpEm8m3WJxiKr21UISuNfve7of6
3jHwV+55Ae2EaEoE3dRPhOrN4YBWe4QmdgjMw+GovA==" fixedSipPort="0" calculatedSipPort="0" configurationD
ataLoaded="0" ppProfileId="0" ppData="" ppnOld="0" ppnOldHex="0x0" />
</GetPPUserResp>
Request :<GetPpProfile id="1" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetRemoteAccess />
Response:<GetRemoteAccessResp enable="1" />
    Insufficient permission!
    Unhandled socket event!
Request :<Subscribe ><e cmd="On" eventType="RFPcfnf" rfpId="-1" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="RFPState" rfpId="-1" /></Subscribe>
Response:<SubscribeResp />
Request :<Subscribe ><e cmd="On" eventType="RFPDetails" rfpId="-1" /></Subscribe>
Response:<SubscribeResp />
Request :<GetRFP id="0" maxRecords="16" withState="1" withDetails="1" />
Response:<GetRFPResp>
    <rfp id="1" ethAddr="00:30:42:0C:BE:9E" dectOn="0" conferenceChannels="0" wlanOn="0" licenseRfp="1"
name="RFP 32 - lxx" hierarchy1="Bei mir rechts" hierarchy2="" hierarchy3="" hierarchy4="" rpn="1" pa
gingArea="0" preferredSync="0" reflectiveEnv="0" site="1" cluster="1" x="0" y="0" hwType="RFP 32" hw
TypeLocked="1" wlanAntenna="0" wlanAntennaDiv="0" wlanHighThroughput="0" wlanChannel="0" wlanProfile
="1" wlanPower="100" connected="0" ipAddr="0.0.0.0" newSoftwareRequest="0" dectRunning="0" wlanRunni
ng="0" wlanLinkNok="0" ommRunning="0" ommStbRunning="0" hasWlan="0" hasEncryption="0" hasAdvancedFea
tures="0" syncState="Inactive" brandingMismatch="0" versionMismatch="0" stbMismatch="0" nSyncRels="0
" swVersion="" wlanChannelUsed="0" wlanHighThroughputTypeUsed="None" wlanPowerUsed="0" />
    <rfp id="2" ethAddr="00:30:42:0A:CD:B0" dectOn="0" conferenceChannels="0" wlanOn="0" licenseRfp="0"
name="License RFP 3" hierarchy1="" hierarchy2="" hierarchy3="" hierarchy4="" rpn="-1" pagingArea="-1
" preferredSync="0" reflectiveEnv="0" site="1" cluster="1" x="0" y="0" hwType="unknown" hwTypeLocked
="0" wlanAntenna="0" wlanAntennaDiv="1" wlanHighThroughput="0" wlanChannel="0" wlanProfile="1" wlanP
ower="100" connected="0" ipAddr="0.0.0.0" newSoftwareRequest="0" dectRunning="0" wlanRunning="0" wla
nLinkNok="0" ommRunning="0" ommStbRunning="0" hasWlan="0" hasEncryption="0" hasAdvancedFeatures="0"
syncState="Inactive" brandingMismatch="0" versionMismatch="0" stbMismatch="0" nSyncRels="0" swVersio

```

```

n="" wlanChannelUsed="0" wlanHighThroughputTypeUsed="None" wlanPowerUsed="0" />
<rfp id="3" ethAddr="00:30:42:11:11:11" dectOn="0" conferenceChannels="0" wlanOn="0" licenseRfp="0"
name="License RFP 3" hierarchy1="" hierarchy2="" hierarchy3="" hierarchy4="" rpn="-1" pagingArea="-1
" preferredSync="0" reflectiveEnv="0" site="1" cluster="1" x="0" y="0" hwType="unknown" hwTypeLocked
="0" wlanAntenna="1" wlanAntennaDiv="1" wlanHighThroughput="0" wlanChannel="1" wlanProfile="1" wlanP
ower="100" connected="0" ipAddr="0.0.0.0" newSoftwareRequest="0" dectRunning="0" wlanRunning="0" wla
nLinkNok="0" ommRunning="0" ommStbRunning="0" hasWlan="0" hasEncryption="0" hasAdvancedFeatures="0"
syncState="Inactive" brandingMismatch="0" versionMismatch="0" stbMismatch="0" nSyncRels="0" swVersio
n="" wlanChannelUsed="0" wlanHighThroughputTypeUsed="None" wlanPowerUsed="0" />
<rfp id="4" ethAddr="00:30:42:12:6A:B1" dectOn="1" conferenceChannels="1" wlanOn="1" licenseRfp="1"
name="RFP43 - 169" hierarchy1="Bei mir links" hierarchy2="" hierarchy3="" hierarchy4="" rpn="1" pagi
ngArea="0" preferredSync="0" reflectiveEnv="0" site="1" cluster="1" x="0" y="0" hwType="RFP 43" hwTy
peLocked="1" wlanAntenna="1" wlanAntennaDiv="0" wlanHighThroughput="0" wlanChannel="10" wlanProfile=
"2" wlanPower="100" connected="1" ipAddr="10.103.35.169" newSoftwareRequest="0" dectRunning="0" wlan
Running="1" wlanLinkNok="0" ommRunning="0" ommStbRunning="0" hasWlan="1" hasEncryption="1" hasAdvanc
edFeatures="1" syncState="NotSynced" brandingMismatch="0" versionMismatch="0" stbMismatch="0" nSyncR
els="0" swVersion="SIP-DECT 6.1-alpha (DIAG) private patch from ckrohne" wlanChannelUsed="0" wlanHig
hThroughputTypeUsed="None" wlanPowerUsed="0" radioType="ConfigurableTX" outdoorType="0" hasFreqShift
="1" />
<rfp id="5" ethAddr="00:30:42:17:74:DA" dectOn="0" conferenceChannels="0" wlanOn="0" licenseRfp="1"
name="RFP 35 - 168" hierarchy1="Bei mir rechts" hierarchy2="" hierarchy3="" hierarchy4="" rpn="2" pa
gingArea="0" preferredSync="0" reflectiveEnv="0" site="1" cluster="1" x="0" y="0" hwType="RFP 35" hw
TypeLocked="1" wlanAntenna="1" wlanAntennaDiv="0" wlanHighThroughput="0" wlanChannel="0" wlanProfile
="1" wlanPower="100" connected="1" ipAddr="10.103.35.168" newSoftwareRequest="0" dectRunning="0" wla
nRunning="0" wlanLinkNok="0" ommRunning="0" ommStbRunning="0" hasWlan="0" hasEncryption="1" hasAdvan
cedFeatures="1" syncState="Inactive" brandingMismatch="0" versionMismatch="0" stbMismatch="0" nSyncR
els="0" swVersion="SIP-DECT 6.1-alpha (DIAG) private patch from ckrohne" wlanChannelUsed="0" wlanHig
hThroughputTypeUsed="None" wlanPowerUsed="0" radioType="ConfigurableTX" outdoorType="0" hasFreqShift
="1" />
</GetRFPResp>
Request :<GetPpProfile id="2" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetEULAConfirm />
Response:<GetEULAConfirmResp confirm="1" />
Request :<GetRFP id="6" maxRecords="16" withState="1" withDetails="1" />
Response:<GetRFPResp errCode="ENoEnt" />
Request :<GetPpProfile id="3" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetNetParams />
Response:<GetNetParamsResp>
  <net voiceToS="176" sigToS="176" ttl="32" voiceEthPrio="0" sigEthPrio="6" />
</GetNetParamsResp>
Request :<GetPpProfile id="4" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetPARK />
Response:<GetPARKResp park="1F10187343" initialPARK="31100303464141" />
Request :<GetPpProfile id="5" />
Response:<GetPpProfileResp>
  <ppProfile id="5" name="Doktor" timeStamp="1430207616" ppData="UD_ConfigurationName=ckrohne-5&#10;UD
  _DispLang=de&#10;UD_DispColor=business&#10;" />
</GetPpProfileResp>
Request :<GetDECTEncryption />
Response:<GetDECTEncryptionResp enable="1" />
Request :<GetPpProfile id="6" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetDECTRegDomain />
Response:<GetDECTRegDomainResp regDomain="EMEA" />
Request :<GetPpProfile id="7" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetDECTAuthCode />
Response:<GetDECTAuthCodeResp ac="" />
Request :<GetPpProfile id="8" />

```

```

Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetDevAutoCreate />
Response:<GetDevAutoCreateResp enable="1" />
Request :<GetPpProfile id="9" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetPPLoginVariant />
Response:<GetPPLoginVariantResp login="NUMBER" />
Request :<GetPpProfile id="10" />
Response:<GetPpProfileResp>
    <ppProfile id="10" name="Profile-10" timeStamp="1430207616" ppData="UD_ConfigurationName=ckrohne-10&
        #10;UD_DisplLang=en&#10;UD_DisplFont=large&#10;UD_DisplColor=gray&#10;" />
    </GetPpProfileResp>
Request :<GetPPFirmwareUpdate />
Response:<GetPPFirmwareUpdateResp enable="1" name="internal" />
Request :<GetPpProfile id="11" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetWLANRegDomain />
Response:<GetWLANRegDomainResp regDomain="DE" channels2_4G="1,2,3,4,5,6,7,8,9,10,11,12,13" channels5G="36,40,
    44,48" />
Request :<GetPpProfile id="12" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetWLANRegDomainList />
Response:<GetWLANRegDomainListResp>
    <regDomain name="AE" />
    <regDomain name="AL" />
    <regDomain name="AM" />
    <regDomain name="AN" />
    <regDomain name="AR" />
    <regDomain name="AT" />
    <regDomain name="AU" />
    <regDomain name="AW" />
    <regDomain name="AZ" />
    <regDomain name="BA" />
    <regDomain name="BB" />
    <regDomain name="BD" />
    <regDomain name="BE" />
    <regDomain name="BG" />
    <regDomain name="BH" />
    <regDomain name="BL" />
    <regDomain name="BN" />
    <regDomain name="BO" />
    <regDomain name="BR" />
    <regDomain name="BY" />
    <regDomain name="BZ" />
    <regDomain name="CA" />
    <regDomain name="CH" />
    <regDomain name="CL" />
    <regDomain name="CN" />
    <regDomain name="CO" />
    <regDomain name="CR" />
    <regDomain name="CY" />
    <regDomain name="CZ" />
    <regDomain name="DE" />
    <regDomain name="DK" />
    <regDomain name="DO" />
    <regDomain name="DZ" />
    <regDomain name="EC" />
    <regDomain name="EE" />
    <regDomain name="EG" />
    <regDomain name="ES" />
    <regDomain name="FI" />
    <regDomain name="FR" />

```

```

    <regDomain name="GE" />
    <regDomain name="GB" />
    <regDomain name="GD" />
    <regDomain name="GR" />
    <regDomain name="GL" />
    <regDomain name="GT" />
    <regDomain name="GU" />
    <regDomain name="HN" />
    <regDomain name="HK" />
    <regDomain name="HR" />
    <regDomain name="HT" />
    <regDomain name="HU" />
    <regDomain name="ID" />
    <regDomain name="IE" />
    <regDomain name="IL" />
    <regDomain name="IN" />
    <regDomain name="IS" />
    <regDomain name="IR" />
    <regDomain name="IT" />
    <regDomain name="JM" />
    <regDomain name="JP" />
    <regDomain name="JO" />
    <regDomain name="KE" />
    <regDomain name="KH" />
    <regDomain name="KP" />
    <regDomain name="KR" />
    <regDomain name="KW" />
    <regDomain name="KZ" />
    <regDomain name="LB" />
    <regDomain name="LI" />
    <regDomain name="LK" />
    <regDomain name="LT" />
    <regDomain name="LU" />
    <regDomain name="LV" />
    <regDomain name="MC" />
    <regDomain name="MA" />
    <regDomain name="MO" />
    <regDomain name="MK" />
    <regDomain name="MT" />
Request : <GetPpProfile id="13" />
    <regDomain name="MY" />
    <regDomain name="MX" />
    <regDomain name="NL" />
    <regDomain name="NO" />
    <regDomain name="NP" />
    <regDomain name="NZ" />
    <regDomain name="OM" />
    <regDomain name="PA" />
    <regDomain name="PE" />
    <regDomain name="PG" />
    <regDomain name="PH" />
    <regDomain name="PK" />
    <regDomain name="PL" />
    <regDomain name="PT" />
    <regDomain name="PR" />
    <regDomain name="QA" />
    <regDomain name="RO" />
    <regDomain name="RS" />
    <regDomain name="RU" />
    <regDomain name="RW" />
    <regDomain name="SA" />
    <regDomain name="SE" />

```

```

    <regDomain name="SG" />
    <regDomain name="SI" />
    <regDomain name="SK" />
    <regDomain name="SV" />
    <regDomain name="SY" />
    <regDomain name="TW" />
    <regDomain name="TH" />
    <regDomain name="TT" />
    <regDomain name="TN" />
    <regDomain name="TR" />
    <regDomain name="UA" />
    <regDomain name="US" />
    <regDomain name="UY" />
    <regDomain name="UZ" />
    <regDomain name="VE" />
    <regDomain name="VN" />
    <regDomain name="YE" />
    <regDomain name="ZA" />
    <regDomain name="ZW" />
  </GetWLANRegDomainListResp>
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetUserDataServer />
Response:<GetUserDataServerResp useCommonFileNameOnServer="0">
  <url enable="1" protocol="TFTP" host="10.103.35.14" port="0" path="open_mob/ck/USER_DATA_DIR/FFSIP"
  useCommonCerts="0" username="" password="somhXDuxz9sSN3xC58uzUjCeKKQTOegzlharm6Twg/ltVyGMW50tIiEtTz
  MqXmrFf3NM9oAKUof8K2jG8HPLg==" />
  </GetUserDataServerResp>
Request :<GetPpProfile id="14" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetBasicSIP />
Response:<GetBasicSIPResp proxyServer="172.30.206.9" proxyPort="5060" outboundProxyServer="" outboundProxyPor
  t="5060" regServer="172.30.206.9" regPort="5060" regPeriod="3600" sipRegisterCheckNum="1" transportP
  rot="UDP" gruu="1" />
Request :<GetPpProfile id="15" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetAdvancedSIP />
Response:<GetAdvancedSIPResp mwiSubscription="1" userAgentInfo="1" xAastraId="1" dialTerminator="" callerDete
  rmination="P-Assert-Identity" regFailedRetryTimer="1200" regTimeoutRetryTimer="180" transactionTimer
  ="4000" blacklistTimeout="0" multipleRing="1" semiAttendedTransferMode="Blind" referToWithReplaces="
  1" userAgentCompatibility="0" callRejectStateCodeUsr="486" callRejectStateCodeDev="486" sessionTimer
  ="0" incomingCallTimeout="180" contactMatching="UsernameOnly" />
Request :<GetPpProfile id="16" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetBackupSIP />
Response:<GetBackupSIPResp failoverActive="0" failoverTime="10" secondaryProxyServer="" secondaryProxyPort="5
  060" secondaryRegServer="" secondaryRegPort="5060" secondaryOutboundProxyServer="" secondaryOutbound
  ProxyPort="5060" tertiaryProxyServer="" tertiaryProxyPort="5060" tertiaryRegServer="" tertiaryRegPor
  t="5060" tertiaryOutboundProxyServer="" tertiaryOutboundProxyPort="5060" />
Request :<GetPpProfile id="17" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetRTP />
Response:<GetRTPResp portBase="16320" packetTime="30" silenceSupp="1" receiverPrecedence="0" comfortNoisePktE
  lim="0" singleCodecReplyInSDP="0">
  <codec type="G.722" />
  <codec type="G.711-A-law" />
  <codec type="G.711-u-law" />
  <codec type="G.711-u-law" />
  </GetRTPResp>
Request :<GetPpProfile id="18" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetDTMF />
Response:<GetDTMFResp outOfBand="1" payloadType="101" method="RFC2833" />

```

SIP-DECT OM APPLICATION XML INTERFACE – AAD-384 V6.1

```
Request :<GetPpProfile id="19" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetSecureSIP />
Response:<GetSecureSIPResp keepAliveTimeoutEnable="1" timeout="30" sendSipsOverTLS="1" authenticationTLS="1"
      commonNameValidationTLS="0" />
Request :<GetPpProfile id="20" />
Response:<GetPpProfileResp errCode="ENoEnt" />
Request :<GetSecureSIPCertificate />
Response:<GetSecureSIPCertificateResp privateKeyPassword="" nTrustedCertificates="1" nLocalCertificates="1" n
      PrivateKeys="1">
  <trustedCertificates>
    <certificate key="-----BEGIN CERTIFICATE-----&#10;MIIDuCCAQCgAwIBAgIBCTANBgkqhkiG9w0BAQUFADBpMQswCQ
      YDQQGEwJERTeP&#10;MA0GA1UECBMGMmVybGlUMQ8wDQYDVQQHEWZCZXJsaW4xPDASBgNVBAoTC0Fhc3Ry&#10;YSBhWjIMQww
      CgYDVQQLLEwNURU0xPDASBgNVBAMTC0NBIHdpbnNlMDA5MB4XDTEz&#10;MDUxNzA2MjgyNloXDTE4MDUxNjA2MjgyNlowWTElMAK
      GALUEBHMCREUXDzANBgNV&#10;BAGTBkJlcmxpbjEUMBIGALUEChMLQWFzdHJhIEdtYkgxMDE4DzANBgNVBAsTA1RFTTEV&#10;MBMGA
      1UEAxMMMTAUMTAZLjlmIjIyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB&#10;CgKCAQEAXdTKjyAF9I5ZX6FEOMjmIqfDTlwW
      CKSDDM5QtIYdHCjEiCdvVwLwH6VC&#10;bnx3uDzIxCGgDegFEE7EbC9GDm41jrCLFmntXFBNra/TIVS9SBZ/lxDf7ROTxuoA&#1
      0;Cumb2J6U+WJICBIZaK6kIB/LiCtPO/AhQoqEtYmz9LctC8ihyfy1Wy6xDqgbfP6Y&#10;gVAq6MBENDB6cWTF17sMCy65nKK9i
      G9F4zvDIAai338FhD1lJarZghXoD7LFCvJ&#10;kYHrpGYKGBZBNjCHTA5NwBfK6pu2n2vnX5LaUB/6lY3+pOdIy7FRKBTxPEjo
      8MH2&#10;exIkxQueYhga2ko2bIPgv0RCniZJywIDAQABo3sweTAJBGNVHRMEAJAAMCwGCWCG&#10;SAGG+EIBDQqFh1PcGvUu1
      NMIEdlbmVYXRlZCDBDZXJ0aWZpY2F0ZTAAdBgNVHQ4E&#10;FgQUKeDGLrtQA+2oPALAxmQINo4Gq8wHwYDVR0jBBGwFoAU4TKTh
      b7uV0GOY7Qb&#10;dLQdCtTt9iIwDQYJKoZIhvcNAQEFBQADggEBALGfLbiH0hf8QmFlhGj1gwKZ8Pf+&#10;XjFngVapK2YKNV
      QQOHpED/XDgX1lgBEoVhaTKLbwfTUxEZmsiwkU2BxTWPk92RU&#10;P3LdLlLVpRh90Gbh9STfleNLw/flzoUOXB6ws/GJRPNj8I
      D7aQB2lcsQ+RWaXN5g&#10;YYQNMd2Gd09NCz2twDcUmQBhLbYjrZrdSLlof+QsfFLCFW7ghulbhb3Bj1EzmK/I&#10;ln5igsOp
      MBCInWiP7adQjasgfw2SyDdptqD1xM3GUil9aEjmvNxae+TFK0+gqm&#10;3bX2a6f1GiEPYzq1QcF9mJS8+G2tsomQNV/lcxW
      4TsSg4QWSI7qlEH2ZKqY=&#10;-----END CERTIFICATE-----&#10;" />
  </trustedCertificates>
  <localCertificates>
    <certificate key="-----BEGIN CERTIFICATE-----&#10;MIIDuCCAQCgAwIBAgIBCTANBgkqhkiG9w0BAQUFADBpMQswCQ
      YDQQGEwJERTeP&#10;MA0GA1UECBMGMmVybGlUMQ8wDQYDVQQHEWZCZXJsaW4xPDASBgNVBAoTC0Fhc3Ry&#10;YSBhWjIMQww
      CgYDVQQLLEwNURU0xPDASBgNVBAMTC0NBIHdpbnNlMDA5MB4XDTEz&#10;MDUxNzA2MjgyNloXDTE4MDUxNjA2MjgyNlowWTElMAK
      GALUEBHMCREUXDzANBgNV&#10;BAGTBkJlcmxpbjEUMBIGALUEChMLQWFzdHJhIEdtYkgxMDE4DzANBgNVBAsTA1RFTTEV&#10;MBMGA
      1UEAxMMMTAUMTAZLjlmIjIyMIIBIjANBgkqhkiG9w0BAQEFAAOCAQ8AMIIB&#10;CgKCAQEAXdTKjyAF9I5ZX6FEOMjmIqfDTlwW
      CKSDDM5QtIYdHCjEiCdvVwLwH6VC&#10;bnx3uDzIxCGgDegFEE7EbC9GDm41jrCLFmntXFBNra/TIVS9SBZ/lxDf7ROTxuoA&#1
      0;Cumb2J6U+WJICBIZaK6kIB/LiCtPO/AhQoqEtYmz9LctC8ihyfy1Wy6xDqgbfP6Y&#10;gVAq6MBENDB6cWTF17sMCy65nKK9i
      G9F4zvDIAai338FhD1lJarZghXoD7LFCvJ&#10;kYHrpGYKGBZBNjCHTA5NwBfK6pu2n2vnX5LaUB/6lY3+pOdIy7FRKBTxPEjo
      8MH2&#10;exIkxQueYhga2ko2bIPgv0RCniZJywIDAQABo3sweTAJBGNVHRMEAJAAMCwGCWCG&#10;SAGG+EIBDQqFh1PcGvUu1
      NMIEdlbmVYXRlZCDBDZXJ0aWZpY2F0ZTAAdBgNVHQ4E&#10;FgQUKeDGLrtQA+2oPALAxmQINo4Gq8wHwYDVR0jBBGwFoAU4TKTh
      b7uV0GOY7Qb&#10;dLQdCtTt9iIwDQYJKoZIhvcNAQEFBQADggEBALGfLbiH0hf8QmFlhGj1gwKZ8Pf+&#10;XjFngVapK2YKNV
      QQOHpED/XDgX1lgBEoVhaTKLbwfTUxEZmsiwkU2BxTWPk92RU&#10;P3LdLlLVpRh90Gbh9STfleNLw/flzoUOXB6ws/GJRPNj8I
      D7aQB2lcsQ+RWaXN5g&#10;YYQNMd2Gd09NCz2twDcUmQBhLbYjrZrdSLlof+QsfFLCFW7ghulbhb3Bj1EzmK/I&#10;ln5igsOp
      MBCInWiP7adQjasgfw2SyDdptqD1xM3GUil9aEjmvNxae+TFK0+gqm&#10;3bX2a6f1GiEPYzq1QcF9mJS8+G2tsomQNV/lcxW
      4TsSg4QWSI7qlEH2ZKqY=&#10;-----END CERTIFICATE-----&#10;" />
  </localCertificates>
</GetSecureSIPCertificateResp>
Request :<GetAutoDBBackup />
Response:<GetAutoDBBackupResp>
  <url enable="1" protocol="TFTP" host="10.103.35.14" port="0" path="open_mob/ck/PC/FFSIP/OMM-6.0-Open
      SER" useCommonCerts="0" username="" password="VgQOfBa400aitYBm7q8IkWKQjaW01w2Q4GGHEOuirjGgfPXB/L3n1
      33gL6YcnLI0ELrkuZNVFW9Y5qBoGS+Q==" />
</GetAutoDBBackupResp>
Request :<GetIMA />
Response:<GetIMAResponse enable="1">
  <url enable="0" protocol="TFTP" host="server" port="0" path="dir" useCommonCerts="0" username="" pas
      sword="M91/bQD2Y0tXNTRUj2QuPbOdZfHnk5sVvmyVALxNqpgqKYV67mqoQPq84FKHucjzZk3nohnNpleIsrW30T0w==" />
</GetIMAResponse>
Request :<GetDECTPagingAreaSize />
Response:<GetDECTPagingAreaSizeResp size="256" />
Request :<GetSNMP />
Response:<GetSNMPResp readCommunity="" contact="" enableTraps="0" trapCommunity="" trapHostAddr="0.0.0.0" />
Request :<GetFACPrefix />
Response:<GetFACPrefixResp prefix="" />
```

```

Request :<GetFACList />
Response:<GetFACListResp>
    <fac feature="ActivateSubscription" enable="1" fac="1" />
    <fac feature="ActivateWildcard" enable="1" fac="2" />
    <fac feature="DeactivateSubscription" enable="1" fac="3" />
    <fac feature="UserLogin" enable="1" fac="4" />
    <fac feature="UserLogout" enable="1" fac="5" />
    <fac feature="SystemCredentialPasswd" enable="0" fac="" />
    <fac feature="BlindTransfer" enable="0" fac="" />
</GetFACListResp>

Request :<GetAutoDBBackupFileName />
Response:<GetAutoDBBackupFileNameResp fileName="150626_OMM_1F10187343_omm_conf.gz" />

Request :<GetSyslogServer />
Response:<GetSyslogServerResp enable="0" forward="0" ipAddr="10.103.35.20" port="10105" />

Request :<GetFlashMemUsage />
Response:<GetFlashMemUsageResp level="-1" />

Request :<GetDECTSubscriptionMode />
Response:<GetDECTSubscriptionModeResp mode="Off" />

Request :<GetHealthState />
Response:<GetHealthStateResp>
    <health component="license" severity="Warning" reason="License with no redundancy" reasonCode="1" />
    <health component="sync" severity="OK" reason="" reasonCode="0" />
    <health component="standby" severity="OK" reason="" reasonCode="0" />
    <health component="dbTransfer" severity="OK" reason="" reasonCode="0" />
    <health component="download" severity="Warning" reason="Firmware update is in start up phase" reasonCode="36" />
    <health component="rfp" severity="OK" reason="" reasonCode="0" />
    <health component="ima" severity="OK" reason="" reasonCode="0" />
    <health component="umo" severity="OK" reason="" reasonCode="0" />
    <health component="uds" severity="OK" reason="" reasonCode="0" />
    <health component="usbMem" severity="OK" reason="" reasonCode="0" />
    <health component="licenseServer" severity="OK" reason="" reasonCode="0" />
    <health component="video" severity="OK" reason="" reasonCode="0" />
    <health component="bluetooth" severity="OK" reason="" reasonCode="0" />
    <health component="sip" severity="OK" reason="" reasonCode="0" />
    <health component="sipCert" severity="OK" reason="" reasonCode="0" />
    <health component="axiProvisioningCommands" severity="OK" reason="" reasonCode="0" />
    <health component="provisioningServer" severity="OK" reason="" reasonCode="0" />
    <health component="dhcpServer" severity="OK" reason="" reasonCode="0" />
    <health component="userDeviceSync" severity="OK" reason="" reasonCode="0" />
</GetHealthStateResp>

Request :<GetSecureSIPCertificateServerImport />
Response:<GetSecureSIPCertificateServerImportResp trustedCertificates="10-103-35-22-cert.pem" localCertificates="10-103-35-22-cert.pem" privateKeys="10-103-35-22-key.pem">
    <url enable="1" protocol="TFTP" host="ber-rd5014" port="0" path="open_mob/ck" useCommonCerts="0" useRname="" password="YvUWjYmWcEbtPqiox5iQrAXHnUkV6IxRr2G4l8XggYyQfHqCka7Qx7wH0QfAHevfrttjl+vHvnbftQIRWn/pvg==" />
</GetSecureSIPCertificateServerImportResp>

Request :<GetUserDeviceSyncOMM />
Response:<GetUserDeviceSyncOMMResp enable="1" omm1DomainName="10.103.35.33" omm2DomainName="" user="omm" password="P9usyBOL3Uw+4EjCNBzCWUs3ZTHfwLtfIsWgZR98IutgVViBnXLQlCRTIh861dwuhW95iv4J/p+VOS9P3xtpA==" />

Request :<GetLicense />
Response:<GetLicenseResp type="standard" state="noRedundancyLicense" latency="43200" park="1F10187343">
    <violation>
    noViolation</violation>
    <licenseRfp id="5" ethAddr="00:30:42:17:74:DA" connected="1" />
    <licenseRfp id="4" ethAddr="00:30:42:12:6A:B1" connected="1" />
    <licenseRfp id="1" ethAddr="00:30:42:0C:BE:9E" connected="0" />
    <sysLicense key="R5N6E-E3CTP-P51ZA-L6AG8-HLG2U" number="256" systemLicenseVersion="6.1" />
    <msgLicense key="WD1TP-EC9PF-38KN9-XM8P9-8PU2Z" number="10000" messagingLicenseRcvMsgs="1" />
    <locLicense key="1MSAS-WNFWT-W296B-N39K7-1MMFB" number="512" locatingLicense="1" />
</GetLicenseResp>

```



```

Request :<GetLicenseServerList />
Response:<GetLicenseServerListResp serverActive="1" installationId="270037827">
    <licenseServer1 server="10.103.35.22" port="38229" serverStandby="" portStandby="38229" />
    <licenseServer2 server="" port="38229" serverStandby="" portStandby="38229" />
    <licenseServer3 server="" port="38229" serverStandby="" portStandby="38229" />
</GetLicenseServerListResp>
Request :<GetRegistrationTrafficShaping />
Response:<GetRegistrationTrafficShapingResp maxRegistrations="4" timeout="0" renewalTimer="15" spreadRegRenew
="1" />
Request :<GetSuplServ />
Response:<GetSuplServResp callForwDiv="0" locLineHndlg="1" callTransferByHook="0" uriSeparator="0" reRegister
AfterFailOver="0" ringingOnHold="1" transferByHook6xxd="1" releaseInfoTimerActiveCall="5" releaseInf
oTimerHoldCall="5" releaseInfoTimerFailedCall="5" />
Request :<GetConferenceServerSIP />
Response:<GetConferenceServerSIPResp conferenceServerType="Integrated" conferenceServerURI="" />
Request :<GetSysVoiceboxNum />
Response:<GetSysVoiceboxNumResp voiceboxNum="Gewürzgurkengläser" />
Request :<GetUserMonitoring />
Response:<GetUserMonitoringResp locatingEscalation="0" startupDelay="600" escalationDelay="180" activityTimeo
utl="43200" activityTimeout2="1800" batteryThresholdValue="20" />
Request :<GetPPFirmwareUpdateOverview />
Response:<GetPPFirmwareUpdateOverviewResp state="startup" />
Request :<GetBluetoothGlobalSettings />
Response:<GetBluetoothGlobalSettingsResp active="0" />
Request :<GetSysToneScheme />
Response:<GetSysToneSchemeResp toneScheme="DE" />
Request :<GetRFPCCapture />
Response:<GetRFPCCaptureResp enable="0" />
Request :<GetDECTPpSettings />
Response:<GetDECTPpSettingsResp dialByNumberOnly="0">
    <bootTextHeadline enable="0" text="" />
    <bootTextStartup enable="0" text="" />
</GetDECTPpSettingsResp>
Request :<GetAdditionalSettings />
Response:<GetAdditionalSettingsResp truncatePpUserName="1" />
Request :<GetSecureSIPCertificateServerImport />
Response:<GetSecureSIPCertificateServerImportResp trustedCertificates="10-103-35-22-cert.pem" localCertificat
es="10-103-35-22-cert.pem" privateKeys="10-103-35-22-key.pem">
    <url enable="1" protocol="TFTP" host="ber-rd5014" port="0" path="open_mob/ck" useCommonCerts="0" use
rname="" password="Maw9PIWLSGCN6kqDACFbgasAghnQF4ip95t7DuYIEqYCNzCC/KTIW3vkPW41zoxNpz+qpgMdJLJB3NpaB
UWCGw==" />
</GetSecureSIPCertificateServerImportResp>
Request :<GetRestrictedSubscriptionDuration />
Response:<GetRestrictedSubscriptionDurationResp restrictedSubscrDur="0" />
Request :<GetSpecialBranding />
Response:<GetSpecialBrandingResp key="">
    <url enable="0" protocol="HTTPS" host="" port="0" path="" useCommonCerts="0" username="" password="u
/WQzUaPl9E3PtXNzAXJP58ZjRdngvFMA8s6l1DnB9peSxAl3lwoWXG8Bvr+27H/fXJHbpzjJFeJxYtozvn+A==" />
</GetSpecialBrandingResp>
Request :<GetConfigURL />
Response:<GetConfigURLResp>
    <url enable="0" protocol="TFTP" host="10.103.35.14" port="0" path="open_mob/xyy/USER_DATA_DIR/FFSIP
" privateKeyPassword="" sslMethod="Auto" validateCerts="0" validateExpires="0" validateHostName="0"
allowNonConfTrustCerts="0" importCerts="0" nTrustedCertificates="0" nLocalCertificates="0" nPrivateK
eys="0" />
</GetConfigURLResp>
Request :<GetUsedConfigURL />
Response:<GetUsedConfigURLResp>
    <url enable="0" protocol="None" host="" port="0" path="" sslMethod="Auto" validateCerts="0" validate
Expires="0" validateHostName="0" allowNonConfTrustCerts="0" importCerts="0" nTrustedCertificates="0"
nLocalCertificates="0" nPrivateKeys="0" />
</GetUsedConfigURLResp>

```

```

Request :<GetSystemCredentials />
Response:<GetSystemCredentialsResp username="omm" password="Di4JPhQwvOfVkBi7r61kl2GEj/ich6RNe6jQSqDzrAX20vIod
      McXFsnvKqltn2lxd4vbyEM+6kykYuR448v8jQ==" />
Request :<GetSystemProvUpdTrig />
Response:<GetSystemProvUpdTrigResp enable="0" hour="0" minute="0" />
Request :<GetPortRangeSIP />
Response:<GetPortRangeSIPResp>
      <userUdpTcp startPort="5060" endPort="5060" />
      <userTls startPort="5061" endPort="5061" />
      <confRoomUdpTcp startPort="5062" endPort="5062" />
      <confRoomTls startPort="5063" endPort="5063" />
</GetPortRangeSIPResp>
Request :<GetIntercomCallHandlingSIP />
Response:<GetIntercomCallHandlingSIPResp autoAnswer="0" microphoneMute="0" warningTone="0" allowBargeIn="0" p
      ushToTalkPrefix="" />
Request :<GetCoreDumpURL />
Response:<GetCoreDumpURLResp>
      <url enable="0" protocol="FTPS" host="10.103.35.178" port="0" path="" username="" password="LNtTL55a
      sfh0ExTfPPYnoDBQ00ylPovLsIbJMA3F+M1vbHaM6mF2NeZ5HpUD40bMyWZGrcMoGNZAX3/vyK10jbxvqNrtmYa4WMsNN+LYQVlk
      0xv6ddakG0HndRpbEf4X0ZjyeAyxWHrbdOtFK+BOc4PGrA/8TxUHsPqgNvCw31A=" />
</GetCoreDumpURLResp>
Request :<GetOMMCertificate />
Response:<GetOMMCertificateResp nLocalCertificates="1" nPrivateKeys="1" privateKeyPassword="Go04evn47mK4/Ut4
      +l8gf8XM5Lr7le0Gu044aC+/fAMf1AnXBmAE0CiAPBMX1lWgERU/XfftNdsJC0NeazjSA==" />
Request :<GetPPFirmwareURL />
Response:<GetPPFirmwareURLResp>
      <url enable="0" protocol="None" host="" port="0" path="" useCommonCerts="0" username="" password="oa
      8uy68XvewthX02JPO+hTH5DCbFPKaQZpAF2x8C9azPi2OhzWocSdRt7JT3gsG8B8N6PTuSgpzMyz+fifsZA==" />
</GetPPFirmwareURLResp>
Request :<GetPreserveUserDeviceRelation />
Response:<GetPreserveUserDeviceRelationResp enable="0" />
Request :<GetSecurePROVCertificateServerImport />
Response:<GetSecurePROVCertificateServerImportResp trustedCertificates="10-103-35-22-cert.pem" localCertifica
      tes="10-103-35-22-cert.pem" privateKeys="10-103-35-22-key.pem">
      <url enable="0" protocol="TFTP" host="ber-rd5014" port="0" path="open_mob/ck" useCommonCerts="0" use
      rname="" password="yhliQTwUpjeYSV56DF18w4SbSAF9T+XNPgjlLg2LtG60MamkhFs8XnJQ5Wxf0BYk3tpdtU6E2/P0kKeqt
      1YaGQ==" />
</GetSecurePROVCertificateServerImportResp>
Request :<GetSecureOMMCertificateServerImport />
Response:<GetSecureOMMCertificateServerImportResp localCertificates="10-103-35-22-cert.pem" privateKeys="10-1
      03-35-22-key.pem">
      <url enable="1" protocol="TFTP" host="ber-rd5014" port="0" path="open_mob/ck" useCommonCerts="0" use
      rname="" password="N+bvDZAZkKiaG/pAMqg4eDam4OewuthwjsNSNWizeM/JNQDYTBs4Q09NvurinTkuljxxD9ankSc068vEt
      axYFQ==" />
</GetSecureOMMCertificateServerImportResp>
Request :<GetRemoteSystemDump />
Response:<GetRemoteSystemDumpResp enable="0" hour="0" minute="0">
      <url enable="0" protocol="TFTP" host="" port="0" path="" useCommonCerts="0" username="" password="LA
      D5PJMunom2cD+0AA2ecMZ6ph6o9NT2oG1P0K/V188nr6TU7kmc/AZmuYN38h8GOIPKT+fKNYe5aNGl6ImgEQ==" />
</GetRemoteSystemDumpResp>
Request :<GetSARI />
Response:<GetSARIResp sari="1F10187342" />
Request :<GetUserDeviceSyncOMM />
Response:<GetUserDeviceSyncOMMResp enable="1" omm1DomainName="10.103.35.33" omm2DomainName="" user="omm" pass
      word="uPynoq7w8a4gJ49gStmhd/kC+dhc9Ou41GQuAAVTBr9hnKVTdTHMlEQhIbb78EGwU6Y0BAv79hyLT57Hn4QH3A=" />
Request :<GetDECTPowerLimit />
Response:<GetDECTPowerLimitResp enable="0" />
Event :<EventRFPState><rftp id="4" dectRunning="1" /></EventRFPState>
Event :<EventRFPSummary nRFPs="5" idFirst="1" nConnected="2" wrongBrandedRFPs="0" wrongStandbyRFPs="0" wron
      gVersionedRFPs="0" newAvailSWRFPs="0" DecryptedDECTRFPs="0" usbOverloads="0" DECTactivatedRFPs="1" D
      ECTactiveRFPs="1" advancedFeaturesErrorRFPs="0" usedDECTclusters="1" usedPagingAreas="1" WLANactivat
      edRFPs="1" WLANrunningRFPs="1" usedWLANprofiles="1" />

```

```

Event      :<EventEventLogEntry>
            <e id="10" time="2015/06/26 08:06:46.759" level="Event" name="IPL" count="1" msg="IPL : 2 of 5 RFPs
            connected" />
            </EventEventLogEntry>
Event      :<EventRFPState><rftp id="4" syncState="Synced" /></EventRFPState>
Event      :<EventRFPState><rftp id="4" wlanChannelUsed="10" wlanPowerUsed="16" /></EventRFPState>

```

5.6.2 OMP Statistik Start Up

Request:<GetSysStatisticConfig />

Request:<GetSysStatisticConfig />

Response:<GetSysStatisticConfigResp>

```

<sysStatName elemId="105" type="occurrence" name="DLC:Abnormal air connection rel" resolution="hour"
duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="106" type="occurrence" name="CC :Transaction Establish" resolution="hour" durat
ion="1" numOfRecords="168" sum="0" />
<sysStatName elemId="108" type="occurrence" name="CC :PP Not Found" resolution="hour" duration="1" n
umOfRecords="168" sum="0" />
<sysStatName elemId="109" type="occurrence" name="CC :Paging for PP" resolution="hour" duration="1"
numOfRecords="168" sum="6" />
<sysStatName elemId="110" type="occurrence" name="CC :Release from PP" resolution="hour" duration="1
" numOfRecords="168" sum="0" />
<sysStatName elemId="111" type="occurrence" name="CC :Setup rejected" resolution="hour" duration="1"
numOfRecords="168" sum="0" />
<sysStatName elemId="112" type="occurrence" name="MM :PP Location registration" resolution="hour" du
ration="1" numOfRecords="168" sum="6" />
<sysStatName elemId="114" type="occurrence" name="MM :PP Detach" resolution="hour" duration="1" numO
fRecords="168" sum="0" />
<sysStatName elemId="115" type="occurrence" name="MM :PP Subscription" resolution="hour" duration="1
" numOfRecords="168" sum="0" />
<sysStatName elemId="116" type="occurrence" name="MM :PP Subscription failed" resolution="hour" dura
tion="1" numOfRecords="168" sum="0" />
<sysStatName elemId="117" type="occurrence" name="DLC:PP Connection Handover" resolution="hour" dura
tion="1" numOfRecords="168" sum="0" />
<sysStatName elemId="120" type="occurrence" name="SS :Transaction Establish" resolution="hour" durat
ion="1" numOfRecords="168" sum="0" />
<sysStatName elemId="121" type="occurrence" name="COMS:Transaction Establish" resolution="hour" dura
tion="1" numOfRecords="168" sum="0" />
<sysStatName elemId="122" type="occurrence" name="SS :Facility received" resolution="hour" duration=
"1" numOfRecords="168" sum="19" />
<sysStatName elemId="123" type="occurrence" name="SS :Facility transmitted" resolution="hour" durati
on="1" numOfRecords="168" sum="19" />
<sysStatName elemId="124" type="occurrence" name="LCE:Link to PP established" resolution="hour" dura
tion="1" numOfRecords="168" sum="6" />
<sysStatName elemId="125" type="occurrence" name="LCE:Link from PP established" resolution="hour" du
ration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="130" type="occurrence" name="IMA:AXI request count" resolution="hour" duration=
"1" numOfRecords="168" sum="7" />
<sysStatName elemId="131" type="occurrence" name="IMA:AXI event count" resolution="hour" duration="1
" numOfRecords="168" sum="3" />
<sysStatName elemId="135" type="occurrence" name="IMA:Msg. sent count" resolution="hour" duration="1
" numOfRecords="168" sum="0" />
<sysStatName elemId="136" type="occurrence" name="IMA:Msg. received count" resolution="hour" duratio
n="1" numOfRecords="168" sum="0" />
<sysStatName elemId="137" type="occurrence" name="IMA:Msg. remote deleted count" resolution="hour" d
uration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="138" type="occurrence" name="IMA:Msg. send/del. retry count" resolution="hour"
duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="139" type="occurrence" name="IMA:Msg. send/del. fail count" resolution="hour" d
uration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="140" type="occurrence" name="IMA:Msg. conf. received count" resolution="hour" d
uration="1" numOfRecords="168" sum="0" />

```

```

<sysStatName elemId="141" type="occurrence" name="IMA:Send prio &apos;info&apos; count" resolution="
hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="142" type="occurrence" name="IMA:Send prio &apos;low&apos; count" resolution="h
our" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="143" type="occurrence" name="IMA:Send prio &apos;normal&apos; count" resolution
="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="144" type="occurrence" name="IMA:Send prio &apos;high&apos; count" resolution="
hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="145" type="occurrence" name="IMA:Send prio &apos;emergency&apos; count" resolut
ion="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="146" type="occurrence" name="IMA:Send prio &apos;locating&apos; count" resoluti
on="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="147" type="occurrence" name="IMA:Mail sent count" resolution="hour" duration="1
" numOfRecords="168" sum="0" />
<sysStatName elemId="148" type="occurrence" name="IMA:Mail received count" resolution="hour" duratio
n="1" numOfRecords="168" sum="0" />
<sysStatName elemId="149" type="occurrence" name="IMA:Mail remote deleted count" resolution="hour" d
uration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="150" type="occurrence" name="IMA:Mail send/del. retry count" resolution="hour"
duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="151" type="occurrence" name="IMA:Mail send/del. fail count" resolution="hour" d
uration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="158" type="occurrence" name="IMA:Alarm trigger rcvd count" resolution="hour" du
ration="1" numOfRecords="168" sum="3" />
<sysStatName elemId="159" type="occurrence" name="IMA:Alarm escalation count" resolution="hour" dura
tion="1" numOfRecords="168" sum="0" />
<sysStatName elemId="160" type="occurrence" name="IMA:Alarm msg. sent count" resolution="hour" durat
ion="1" numOfRecords="168" sum="0" />
<sysStatName elemId="161" type="occurrence" name="IMA:Alarm msg. conf. count" resolution="hour" dura
tion="1" numOfRecords="168" sum="0" />
<sysStatName elemId="170" type="occurrence" name="+STB:Switch to standby OMM" resolution="hour" dura
tion="1" numOfRecords="1" sum="0" />
<sysStatName elemId="171" type="occurrence" name="+STB:Loss of standby OMM" resolution="hour" durati
on="1" numOfRecords="168" sum="0" />
<sysStatName elemId="172" type="occurrence" name="+STB:Update of standby OMM" resolution="hour" dura
tion="1" numOfRecords="168" sum="0" />
<sysStatName elemId="180" type="occurrence" name="+ADB:Done database backups" resolution="hour" dura
tion="1" numOfRecords="168" sum="1" />
<sysStatName elemId="181" type="occurrence" name="+ADB:Failed database backups" resolution="hour" du
ration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="182" type="occurrence" name="+ADB:Done database restores" resolution="hour" dur
ation="1" numOfRecords="1" sum="0" />
<sysStatName elemId="183" type="occurrence" name="+ADB:Done identical database restores" resolution=
"hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="184" type="occurrence" name="+ADB:Failed database restores" resolution="hour" d
uration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="190" type="occurrence" name="EPR:Done user logins" resolution="hour" duration="
1" numOfRecords="168" sum="0" />
<sysStatName elemId="191" type="occurrence" name="EPR:User login requests" resolution="hour" duratio
n="1" numOfRecords="168" sum="0" />
<sysStatName elemId="192" type="occurrence" name="EPR:Done user logouts" resolution="hour" duration=
"1" numOfRecords="168" sum="0" />
<sysStatName elemId="193" type="occurrence" name="EPR:User logout requests" resolution="hour" durati
on="1" numOfRecords="168" sum="0" />
<sysStatName elemId="194" type="occurrence" name="EPR:Done file imports" resolution="hour" duration=
"1" numOfRecords="168" sum="7" />
<sysStatName elemId="195" type="occurrence" name="EPR:Failed file imports" resolution="hour" duratio
n="1" numOfRecords="168" sum="0" />
<sysStatName elemId="200" type="occurrence" name="Messages: failed messages" resolution="hour" durat
ion="1" numOfRecords="168" sum="0" />
<sysStatName elemId="201" type="occurrence" name="Messages: queued before start of send attempt" res
olution="hour" duration="1" numOfRecords="168" sum="21" />

```

```

<sysStatName elemId="202" type="occurrence" name="Messages: immediate start of send attempt" resolution="hour" duration="1" numOfRecords="168" sum="4" />
<sysStatName elemId="203" type="occurrence" name="Messages: canceled" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="204" type="occurrence" name="Messages: deleted" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="205" type="occurrence" name="Messages: undeliverable, removed from queue after 2h" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="225" type="occurrence" name="XML:Maximal parallel sessions" resolution="hour" duration="1" numOfRecords="168" sum="6" />
<sysStatName elemId="226" type="occurrence" name="XML:SIP notify requests" resolution="hour" duration="1" numOfRecords="168" sum="18" />
<sysStatName elemId="227" type="occurrence" name="XML:Caller list requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="228" type="occurrence" name="XML:Redial list requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="229" type="occurrence" name="XML:User presence requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="230" type="occurrence" name="XML:Server application menu requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="232" type="occurrence" name="XML:Corporate directory application-1 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="233" type="occurrence" name="XML:Corporate directory application-2 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="234" type="occurrence" name="XML:Corporate directory application-3 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="235" type="occurrence" name="XML:Corporate directory application-4 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="236" type="occurrence" name="XML:Corporate directory application-5 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="237" type="occurrence" name="XML:Configured application-1 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="238" type="occurrence" name="XML:Configured application-2 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="239" type="occurrence" name="XML:Configured application-3 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="240" type="occurrence" name="XML:Configured application-4 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="241" type="occurrence" name="XML:Configured application-5 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="242" type="occurrence" name="XML:Configured application-6 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="243" type="occurrence" name="XML:Configured application-7 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="244" type="occurrence" name="XML:Configured application-8 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="245" type="occurrence" name="XML:Configured application-9 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="246" type="occurrence" name="XML:Configured application-10 requests" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="247" type="occurrence" name="XML:Requested documents" resolution="hour" duration="1" numOfRecords="168" sum="6" />
<sysStatName elemId="248" type="occurrence" name="XML:Application errors" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="249" type="occurrence" name="XML:Document errors" resolution="hour" duration="1" numOfRecords="168" sum="6" />
<sysStatName elemId="250" type="occurrence" name="XML:Parse errors" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="252" type="occurrence" name="G729: connections per hour" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="253" type="occurrence" name="G729: missed licenses per hour" resolution="hour" duration="1" numOfRecords="168" sum="0" />

```

```

<sysStatName elemId="254" type="occurrence" name="G729: aborted conn. per hour" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="256" type="occurrence" name="UMO:AXI request count" resolution="hour" duration="1" numOfRecords="168" sum="7" />
<sysStatName elemId="257" type="occurrence" name="UMO:AXI event count" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="261" type="occurrence" name="UMO:Active pagings" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="262" type="occurrence" name="UMO:Alarm trigger OK" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="263" type="occurrence" name="UMO:Alarm trigger Warning" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="264" type="occurrence" name="UMO:Alarm trigger Error" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="265" type="occurrence" name="UMO:Alarm trigger Loc-Error" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="268" type="occurrence" name="SYN:RFP synchronisation losts" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="269" type="occurrence" name="SYN:Active rels changes to 1" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="270" type="occurrence" name="SYN:RFP offset jumps" resolution="hour" duration="1" numOfRecords="168" sum="0" />
<sysStatName elemId="118" type="minmax" name="DLC: Call duration" resolution="hour" duration="1" numOfRecords="168" unit="sec" numOfFreqDistUnits="18" />
<sysStatName elemId="119" type="minmax" name="DLC: Connection duration" resolution="hour" duration="1" numOfRecords="168" unit="sec" numOfFreqDistUnits="18" />
<sysStatName elemId="132" type="minmax" name="IMA:AXI requests pending" resolution="hour" duration="1" numOfRecords="168" unit="count/s" numOfFreqDistUnits="21" />
<sysStatName elemId="133" type="minmax" name="IMA:Msg&apos;s queued" resolution="hour" duration="1" numOfRecords="168" unit="count/s" numOfFreqDistUnits="21" />
<sysStatName elemId="134" type="minmax" name="IMA:Active alarms" resolution="hour" duration="1" numOfRecords="168" unit="count/s" numOfFreqDistUnits="21" />
<sysStatName elemId="152" type="minmax" name="IMA:Send msg. content length" resolution="hour" duration="1" numOfRecords="168" unit="bytes/msg" numOfFreqDistUnits="21" />
<sysStatName elemId="153" type="minmax" name="IMA:Msg. delivery time" resolution="hour" duration="1" numOfRecords="168" unit="s/msg" numOfFreqDistUnits="19" />
<sysStatName elemId="154" type="minmax" name="IMA:Msg. pure delivery time" resolution="hour" duration="1" numOfRecords="168" unit="ms/msg" numOfFreqDistUnits="31" />
<sysStatName elemId="155" type="minmax" name="IMA:Msg. deletion time" resolution="hour" duration="1" numOfRecords="168" unit="s/msg" numOfFreqDistUnits="19" />
<sysStatName elemId="156" type="minmax" name="IMA:Mail delivery time" resolution="hour" duration="1" numOfRecords="168" unit="s/mail" numOfFreqDistUnits="19" />
<sysStatName elemId="157" type="minmax" name="IMA:Mail deletion time" resolution="hour" duration="1" numOfRecords="168" unit="s/mail" numOfFreqDistUnits="19" />
<sysStatName elemId="162" type="minmax" name="IMA:Alarm msg. conf. time" resolution="hour" duration="1" numOfRecords="168" unit="s/alarm" numOfFreqDistUnits="19" />
<sysStatName elemId="163" type="minmax" name="IMA:Alarm lifetime" resolution="hour" duration="1" numOfRecords="168" unit="s/alarm" numOfFreqDistUnits="19" />
<sysStatName elemId="206" type="minmax" name="Messages: delivery time" resolution="hour" duration="1" numOfRecords="168" unit="ms/msg" numOfFreqDistUnits="31" />
<sysStatName elemId="255" type="minmax" name="G729: simult. used licenses per hour" resolution="hour" duration="1" numOfRecords="168" unit="licenses" numOfFreqDistUnits="11" />
<sysStatName elemId="258" type="minmax" name="UMO:AXI requests pending" resolution="hour" duration="1" numOfRecords="168" unit="count/s" numOfFreqDistUnits="10" />
<sysStatName elemId="259" type="minmax" name="UMO:Monitored users (passive)" resolution="hour" duration="1" numOfRecords="168" unit="count" numOfFreqDistUnits="10" />
<sysStatName elemId="260" type="minmax" name="UMO:Monitored users (active)" resolution="hour" duration="1" numOfRecords="168" unit="count" numOfFreqDistUnits="10" />
<sysStatName elemId="266" type="minmax" name="SYN:RFP startup time" resolution="hour" duration="1" numOfRecords="1" unit="sec" numOfFreqDistUnits="15" />
<sysStatName elemId="267" type="minmax" name="SYN:Cluster startup time" resolution="day" duration="1" numOfRecords="1" unit="sec" numOfFreqDistUnits="15" />

```



```

    <enumeration value="EEncryptNotAllowed"/>
    <enumeration value="EExist"/>
    <enumeration value="EFailed"/>
    <enumeration value="EForbidden"/>
    <enumeration value="EInProgress"/>
    <enumeration value="EInval"/>
    <enumeration value="EInvalidChars"/>
    <enumeration value="ELicense"/>
    <enumeration value="ELicenseFile"/>
    <enumeration value="ELicenseWrongInstallId"/>
    <enumeration value="EMissing"/>
    <enumeration value="ENoEnt"/>
    <enumeration value="ENoMem"/>
    <enumeration value="EPerm"/>
    <enumeration value="EPwEmpty"/>
    <enumeration value="EPwUnchanged"/>
    <enumeration value="EPwSimilarToHost"/>
    <enumeration value="EPwSimilarToName"/>
    <enumeration value="EPwTooManySimilarChars"/>
    <enumeration value="EPwTooShort"/>
    <enumeration value="EPwTooSimilar"/>
    <enumeration value="EPwTooWeak"/>
    <enumeration value="ETooLong"/>
    <enumeration value="EWlanRegDomainInvalid"/>
  </restriction>
</simpleType>

<simpleType name="ProtocolType">
  <restriction base="string">
    <enumeration value="FTP"> </enumeration>
    <enumeration value="FTPS"> </enumeration>
    <enumeration value="HTTP"> </enumeration>
    <enumeration value="HTTPS"> </enumeration>
    <enumeration value="TFTP"> </enumeration>
    <enumeration value="SFTP"> </enumeration>
    <enumeration value="SCP"> </enumeration>
    <enumeration value="None"> </enumeration>
  </restriction>
</simpleType>

<simpleType name=" SSLMethodType ">
  <restriction base="string">
    <enumeration value="Auto"> </enumeration>
    <enumeration value="TLS1.0"> </enumeration>
    <enumeration value="TLS1.1"> </enumeration>
    <enumeration value="TLS1.2"> </enumeration>
  </restriction>
</simpleType>

<complexType name="URLType">
  <attribute name="enable" type="boolean" use="optional"/>
  <attribute name="useCommonCerts" type="boolean" use="optional"/>
  <attribute name="protocol" type="tns:ProtocolType" use=" optional "/>
  <attribute name="host" type="string" use=" optional "/>
  <attribute name="port" type="integer" use=" optional "/>
  <attribute name="path" type="string" use=" optional "/>
  <attribute name="username" type="string" use="optional"/>
  <attribute name="password" type="string" use="optional"/>
  <attribute name="sslMethod" type="tns:SSLMethodType" use="optional"/>
  <attribute name="privateKeyPassword" type="string" use="optional"/>
  <attribute name="validateCerts" type="boolean" use="optional"/>
  <attribute name="validateExpires" type="boolean" use="optional"/>

```

```

<attribute name="validateHostName" type="boolean" use="optional"/>
<attribute name="allowNonConfTrustCerts" type="boolean" use="optional"/>
<attribute name="importCerts" type="boolean" use="optional"/>
<attribute name="nTrustedCertificates" type="integer" use=" optional "/>
<attribute name="nLocalCertificates" type="integer" use=" optional "/>
<attribute name="nPrivateKeys" type="integer" use=" optional "/>
<attribute name="trustedCertificates" type="tns:SetSecureSIPListType"/>
<attribute name="localCertificates" type="tns:SetSecureSIPListType"/>
<attribute name="privateKey" type="tns:SetSecureSIPListType"/>
<attribute name="privateKeyPassword" type="string" use="optional"/>
</complexType>

<complexType name="ServerType">
  <complexContent>
    <attribute name="url" type="tns:URLType"/>
    <attribute name="enable" type="boolean"/>
  </complexContent>
</complexType>

<simpleType name="PermissionType">
  <restriction base="string">
    <enumeration value="AllCnfWrite"/>
    <enumeration value="AllCnfRead"/>
    <enumeration value="InfoMessaging"/>
    <enumeration value="Messaging"/>
    <enumeration value="Alerting"/>
    <enumeration value="LocatingAlert"/>
    <enumeration value="Locating"/>
    <enumeration value="Monitoring"/>
    <enumeration value="Conferencing"/>
    <enumeration value="Video"/>
  </restriction>
</simpleType>

<complexType name="DateTimeType">
  <attribute name="year" type="integer" use="required"/>
  <attribute name="month" type="integer" use="required"/>
  <attribute name="day" type="integer" use="required"/>
  <attribute name="hour" type="integer" use="required"/>
  <attribute name="minute" type="integer" use="required"/>
  <attribute name="second" type="integer" use="optional"/>
</complexType>

<simpleType name="ComponentType">
  <restriction base="string">
    <enumeration value="sync"/>
    <enumeration value="standby"/>
    <enumeration value="autoDB"/>
    <enumeration value="dbTransfer"/>
    <enumeration value="download"/>
    <enumeration value="rfp"/>
    <enumeration value="license"/>
    <enumeration value="ima"/>
    <enumeration value="licenseG729"/>
    <enumeration value="umo"/>
    <enumeration value="uds"/>
    <enumeration value="usbMem"/>
    <enumeration value="licenseServer"/>
    <enumeration value="video"/>
    <enumeration value="bluetooth"/>
    <enumeration value="sip"/>
    <enumeration value="sipCert"/>
  </restriction>
</simpleType>

```

```

        <enumeration value="axiProvisioningCommands"/>
        <enumeration value="provisioningServer"/>
        <enumeration value="dhcpServer"/>
        <enumeration value="userDeviceSync"/>
    </restriction>
</simpleType>

<simpleType name="ReasonType">
    <restriction base="string">
        <enumeration value=""/>
        <enumeration value="License with no redundancy"/>
        <enumeration value="License blocked"/>
        <enumeration value="Inactive license"/>
        <enumeration value="No license"/>
        <enumeration value="Split cluster"/>
        <enumeration value="Connection broken"/>
        <enumeration value="Manual import"/>
        <enumeration value="Manual export"/>
        <enumeration value="Automatic export"/>
        <enumeration value="Firmware not found"/>
        <enumeration value="Firmware update is in start up phase"/>
        <enumeration value="Branding mismatch"/>
        <enumeration value="No encryption"/>
        <enumeration value="Protocol mismatch"/>
        <enumeration value="Cannot connect / lost connection to OMM"/>
        <enumeration value="Cannot receive IMA configuration file"/>
        <enumeration value="High message queue level"/>
        <enumeration value="Cannot send messages"/>
        <enumeration value="Cannot delete messages"/>
        <enumeration value="Cannot send e-mails"/>
        <enumeration value="Cannot receive e-mails"/>
        <enumeration value="Cannot delete e-mails"/>
        <enumeration value="Cannot receive RSS feed"/>
        <enumeration value="Wrong standby configuration"/>
        <enumeration value="License latency period expired"/>
        <enumeration value="All channels in use"/>
        <enumeration value="User monitoring is in start up phase"/>
        <enumeration value="Too many users for user monitoring configured"/>
        <enumeration value="User monitoring loses OMM AXI connection"/>
        <enumeration value="USB overload"/>
        <enumeration value="An RFP without advanced features is connected to a site, where advanced features are
activated"/>
        <enumeration value="Import of the common user data file from server failed"/>
        <enumeration value="Invalid configuration data in common user data file"/>
        <enumeration value="Import of at least one user data file from server failed"/>
        <enumeration value="Invalid configuration data in at least one user data file"/>
        <enumeration value="USB memory is missing"/>
        <enumeration value="Configured server not started"/>
        <enumeration value="License client not connected"/>
        <enumeration value="License file parse error"/>
        <enumeration value="Received no license from server"/>
        <enumeration value="Missing standby license server(s)"/>
        <enumeration value="The configured installation ID does not fit to the license file"/>
        <enumeration value="An error of a video device was detected, please check the error log for more information"/>
        <enumeration value="A plugged and activated video device gets unplugged"/>
        <enumeration value="An error of a Bluetooth beacon was detected"/>
        <enumeration value="At least one Bluetooth beacon is unplugged"/>
        <enumeration value="HTTP error for SIP secure trusted certificate"/>
        <enumeration value="Imported SIP secure trusted certificate not stored in database"/>
        <enumeration value="File transfer error (wrong configuration?) for SIP secure trusted certificate"/>
        <enumeration value="HTTP error for SIP secure local certificate"/>
        <enumeration value="Imported SIP secure local certificate not stored in database"/>
    </restriction>
</simpleType>

```

```

    <enumeration value="File transfer error (wrong configuration?) for SIP secure trusted certificate"/>
    <enumeration value="HTTP error for SIP secure local certificate"/>
    <enumeration value="Imported SIP secure local certificate not stored in database"/>
    <enumeration value="File transfer error (wrong configuration?) for SIP secure local certificate"/>
    <enumeration value="HTTP error for SIP secure private key"/>
    <enumeration value="Imported SIP secure private key not stored in database"/>
    <enumeration value="File transfer error (wrong configuration?) for SIP secure private key"/>
    <enumeration value="AXI command error(s) during provisioning file processing"/>
    <enumeration value="Access to provisioning server failed"/>
    <enumeration value="Invalid IP address range configuration in DHCP server (range too small for number of RFPs)"/>
    <enumeration value="Cannot connect to external AXI / lost AXI connection"/>
    <enumeration value="Different AXI or OMM version"/>
    <enumeration value="Connection between RFP and PC OMM"/>
    <enumeration value="Different system times"/>
    <enumeration value="User/device synchronization is activated on configured central OMM, too"/>
    <enumeration value="SARI not configured on central OMM"/>
  </restriction>
</simpleType>

<simpleType name="SeverityType">
  <restriction base="string">
    <enumeration value="OK"/>
    <enumeration value="Idle"/>
    <enumeration value="Warning"/>
    <enumeration value="Error"/>
  </restriction>
</simpleType>

<complexType name="HealthType">
  <attribute name="component" type="tns:ComponentType" use="required"/>
  <attribute name="severity" type="tns:SeverityType" use="required"/>
  <attribute name="reason" type="string" use="optional"/>
  <attribute name="reasonCode" type="tns:ReasonType" use="required"/>
  <simpleType>
    <restriction base="integer">
      <minInclusive value="0"/>
      <maxInclusive value="29"/>
    </restriction>
  </simpleType>
</complexType>

<complexType name="GetHealthStateRespType">
  <complexContent>
    <extension base="tns:RespType">
      <sequence>
        <attribute name="health" type="tns:HealthType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<simpleType name="PPUserCallStateType">
  <restriction base="string">
    <enumeration value="ringing"/>
    <enumeration value="calling"/>
    <enumeration value="paging"/>
    <enumeration value="connected"/>
    <enumeration value="idle"/>
    <enumeration value="none"/>
  </restriction>
</simpleType>

```

```

<simpleType name="PPUserSIPEventType">
  <restriction base="string">
    <enumeration value="sipRegistrationEvent"/>
    <enumeration value="sipRegistrationEndEvent"/>
    <enumeration value="sipNotifyEvent"/>
  </restriction>
</simpleType>

<complexType name="AccountStateType">
  <attribute name="id" use="required">
    <simpleType>
      <restriction base="integer">
        <minInclusive value="0"/>
        <maxInclusive value="2"/>
      </restriction>
    </simpleType>
  </attribute>
  <attribute name="username" type="string" use="required"/>
  <attribute name="state" type="string" use="required"/>
</complexType>

<simpleType name="VersionElementType">
  <restriction base="string">
    <enumeration value="ActivateRemoteSystemDump"/>
    <enumeration value="AddUserToConference"/>
    <enumeration value="CancelMessage"/>
    <enumeration value="ChangeUserInConference"/>
    <enumeration value="ConferenceConfirmation"/>
    <enumeration value="CreateAccount"/>
    <enumeration value="CreateACLEntry"/>
    <enumeration value="CreateAlarmTrigger"/>
    <enumeration value="CreateBluetoothBeacon"/>
    <enumeration value="CreateConference"/>
    <enumeration value="CreateConferenceRoom"/>
    <enumeration value="CreateDigitTreatment"/>
    <enumeration value="CreateFixedPP"/>
    <enumeration value="CreateLDAP"/>
    <enumeration value="CreatePPDev"/>
    <enumeration value="CreatePPUser"/>
    <enumeration value="CreateRFP"/>
    <enumeration value="CreateSite"/>
    <enumeration value="CreateVideoDev"/>
    <enumeration value="CreateWLANProfile"/>
    <enumeration value="CreateXMLApplication"/>
    <enumeration value="DBBackupToUSB"/>
    <enumeration value="DBRestoreFromUSB"/>
    <enumeration value="DeleteAccount"/>
    <enumeration value="DeleteACLEntry"/>
    <enumeration value="DeleteAlarmTrigger"/>
    <enumeration value="DeleteBluetoothBeacon"/>
    <enumeration value="DeleteConference"/>
    <enumeration value="DeleteConferenceRoom"/>
    <enumeration value="DeleteDigitTreatment"/>
    <enumeration value="DeleteEventLogBuffer"/>
    <enumeration value="DeleteUserFromConference"/>
    <enumeration value="DeleteLDAP"/>
    <enumeration value="DeleteMessage"/>
    <enumeration value="DeletePPDev"/>
    <enumeration value="DeletePPUser"/>
    <enumeration value="DeleteRFP"/>
    <enumeration value="DeleteRFPCaptureList"/>
    <enumeration value="DeleteRFPCaptureListElem"/>
  </restriction>

```

```

<enumeration value="DeleteSite"/>
<enumeration value="DeleteVideoDev"/>
<enumeration value="DeleteWLANProfile"/>
<enumeration value="DeleteXMLApplication"/>
<enumeration value="EventAccountCnf"/>
<enumeration value="EventAccountSummary"/>
<enumeration value="EventACLCnf"/>
<enumeration value="EventAdditionalSettingsCnf"/>
<enumeration value="EventAddUserToConference"/>
<enumeration value="EventAdvancedSIPCnf"/>
<enumeration value="EventAlarmCallProgress"/>
<enumeration value="EventAlarmTriggerCnf"/>
<enumeration value="EventAlarmTrigger"/>
<enumeration value="EventAutoDBBackupCnf"/>
<enumeration value="EventAutoDBBackupFileNameCnf"/>
<enumeration value="EventBackupSIPCnf"/>
<enumeration value="EventBasicSIPCnf"/>
<enumeration value="EventBluetoothClientStatistic"/>
<enumeration value="EventBluetoothBeaconCnf"/>
<enumeration value="EventBluetoothBeaconSummary"/>
<enumeration value="EventBluetoothGlobalSettingsCnf"/>
<enumeration value="EventBluetoothSensitivityCnf"/>
<enumeration value="EventChangeUserInConference"/>
<enumeration value="EventConference"/>
<enumeration value="EventConferenceRequest"/>
<enumeration value="EventConferenceRoomCnf"/>
<enumeration value="EventConferenceServerSIPCnf"/>
<enumeration value="EventConfigURLCnf"/>
<enumeration value="EventCoreDumpCnf"/>
<enumeration value="EventCreateConference"/>
<enumeration value="EventDbTransferState"/>
<enumeration value="EventDECTAuthCodeCnf"/>
<enumeration value="EventDECTEncryptionCnf"/>
<enumeration value="EventDECTPagingAreaSizeCnf"/>
<enumeration value="EventDECTPowerLimitCnf"/>
<enumeration value="EventDECTPpSettingsCnf"/>
<enumeration value="EventDECTRegDomainCnf"/>
<enumeration value="EventDECTSubscriptionMode"/>
<enumeration value="EventDeleteConference"/>
<enumeration value="EventDeleteUserFromConference"/>
<enumeration value="EventDevAutoCreateCnf"/>
<enumeration value="EventDigitTreatmentCnf"/>
<enumeration value="EventDTMFCnf"/>
<enumeration value="EventEULAConfirmCnf"/>
<enumeration value="EventEventLogEntry"/>
<enumeration value="EventFACcnf"/>
<enumeration value="EventFACPrefixCnf"/>
<enumeration value="EventFreeConferenceChannels"/>
<enumeration value="EventHealthState"/>
<enumeration value="EventIMACnf"/>
<enumeration value="EventIntercomCallHandlingSIPCnf"/>
<enumeration value="EventLDAPCnf"/>
<enumeration value="EventLicenseCnf"/>
<enumeration value="EventLicensedCodecLines"/>
<enumeration value="EventLicenseFile"/>
<enumeration value="EventLicenseServerListCnf"/>
<enumeration value="EventMessageConfirmation"/>
<enumeration value="EventMessageProgress"/>
<enumeration value="EventMessageQueueEmpty"/>
<enumeration value="EventMessageSend"/>
<enumeration value="EventNetParamsCnf"/>
<enumeration value="EventNTPServerCnf"/>

```

```

<enumeration value="EventOMMCertificateCnf"/>
<enumeration value="EventOMPURLCnf"/>
<enumeration value="EventPARKCnf"/>
<enumeration value="EventPARKFromServerResult"/>
<enumeration value="EventPBXEnrolmentCnf"/>
<enumeration value="EventPermissionChange"/>
<enumeration value="EventPositionHistory"/>
<enumeration value="EventPositionInfo"/>
<enumeration value="EventPositionRequest"/>
<enumeration value="EventPositionTrack"/>
<enumeration value="EventPortRangeSIPCnf"/>
<enumeration value="EventPPDevCnf"/>
<enumeration value="EventPpProfileCnf"/>
<enumeration value="EventPPFirmwareUpdateCnf"/>
<enumeration value="EventPPFirmwareUpdateOverview"/>
<enumeration value="EventPPFirmwareUpdateStatus"/>
<enumeration value="EventPPFirmwareCnf"/>
<enumeration value="EventPPLoginVariantCnf"/>
<enumeration value="EventPPState"/>
<enumeration value="EventPPTransaction"/>
<enumeration value="EventPPUserCnf"/>
<enumeration value="EventPPUserSummary"/>
<enumeration value="EventPreserveUserDeviceRelationCnf"/>
<enumeration value="EventReadyForConferencing"/>
<enumeration value="EventRegistrationTrafficShapingCnf"/>
<enumeration value="EventRemoteAccessCnf"/>
<enumeration value="EventRemoteSystemDumpCnf"/>
<enumeration value="EventRestrictedSubscriptionDurationCnf"/>
<enumeration value="EventRFPCapture"/>
<enumeration value="EventRFPCaptureCnf"/>
<enumeration value="EventRFPCnf"/>
<enumeration value="EventRFPCConnectAttempt"/>
<enumeration value="EventRFPCDetails"/>
<enumeration value="EventRFPIpQuality"/>
<enumeration value="EventRFPMCnf"/>
<enumeration value="EventRFPMediaStreamQuality"/>
<enumeration value="EventRFPState"/>
<enumeration value="EventRFPSummary"/>
<enumeration value="EventRFPsyncQuality"/>
<enumeration value="EventRFPMediaStreamQuality"/>
<enumeration value="EventRFPsyncRel"/>
<enumeration value="EventRTPConferenceStreamChg"/>
<enumeration value="EventRTPCnf"/>
<enumeration value="EventSARICnf"/>
<enumeration value="EventSecureOMMCertificateCnfServerImport"/>
<enumeration value="EventSecurePROVCertificateCnfServerImport"/>
<enumeration value="EventSecureSIPCertificateCnf"/>
<enumeration value="EventSecureSIPCertificateCnfServerImport"/>
<enumeration value="EventSecureSIPCnf"/>
<enumeration value="EventSiteCnf"/>
<enumeration value="EventSiteSummary"/>
<enumeration value="EventSNMPCnf"/>
<enumeration value="EventSoftwareImageCnf"/>
<enumeration value="EventSpecialBrandingCnf"/>
<enumeration value="EventStbStateChange"/>
<enumeration value="EventSuplServCnf"/>
<enumeration value="EventSyslogServerCnf"/>
<enumeration value="EventSystemCredentialsCnf"/>
<enumeration value="EventSystemNameCnf"/>
<enumeration value="EventSystemProvUpdTrigCnf"/>
<enumeration value="EventSysToneSchemeCnf"/>
<enumeration value="EventSysVoiceboxNumCnf"/>

```

```

<enumeration value="EventTimeZoneCnf" />
<enumeration value="EventTimeZoneDetailsCnf" />
<enumeration value="EventTimeZoneList" />
<enumeration value="EventUsedConfigURL" />
<enumeration value="EventUserDataServerCnf" />
<enumeration value="EventUserDeviceSyncOMMCnf" />
<enumeration value="EventUserMonitoringCnf" />
<enumeration value="EventVideoDevCnf" />
<enumeration value="EventVideoDevSummary" />
<enumeration value="EventWLANClient" />
<enumeration value="EventWLANProfileCnf" />
<enumeration value="EventWLANRegDomainCnf" />
<enumeration value="EventXMLApplicationCnf" />
<enumeration value="GenerateHealthStateAlarmTriggers" />
<enumeration value="GetAccount" />
<enumeration value="GetAccountSummary" />
<enumeration value="GetACLEntry" />
<enumeration value="GetActivePPDev" />
<enumeration value="GetAdditionalSettings" />
<enumeration value="GetAdvancedSIP" />
<enumeration value="GetAlarmTrigger" />
<enumeration value="GetAlarmTriggerSummary" />
<enumeration value="GetAutoDBBackup" />
<enumeration value="GetAutoDBBackupFileName" />
<enumeration value="GetBackupSIP" />
<enumeration value="GetBasicSIP" />
<enumeration value="GetBluetoothBeacon" />
<enumeration value="GetBluetoothBeaconSummary" />
<enumeration value="GetBluetoothGlobalSettings" />
<enumeration value="GetBluetoothSensitivity" />
<enumeration value="GetConferenceRoom" />
<enumeration value="GetConferenceServerSIP" />
<enumeration value="GetConfigURL" />
<enumeration value="GetCoreDumpURL" />
<enumeration value="GetDbTransferState" />
<enumeration value="GetDECTAuthCode" />
<enumeration value="GetDECTEncryption" />
<enumeration value="GetDECTPagingAreaSize" />
<enumeration value="GetDECTPowerLimit" />
<enumeration value="GetDECTPpSettings" />
<enumeration value="GetDECTRegDomain" />
<enumeration value="GetDECTSubscriptionMode" />
<enumeration value="GetDevAutoCreate" />
<enumeration value="GetDigitTreatment" />
<enumeration value="GetDigitTreatmentSummary" />
<enumeration value="GetDTMF" />
<enumeration value="GetEULAConfirm" />
<enumeration value="GetEventLogBuffer" />
<enumeration value="GetFACLlist" />
<enumeration value="GetFACPrefix" />
<enumeration value="GetFile" />
<enumeration value="GetFlashMemUsage" />
<enumeration value="GetFreeConferenceChannels" />
<enumeration value="GetG729ChannelsForConference" />
<enumeration value="GetHealthState" />
<enumeration value="GetIMA" />
<enumeration value="GetIntercomCallHandlingSIP" />
<enumeration value="GetLastPPDevAction" />
<enumeration value="GetLDAP" />
<enumeration value="GetLicense" />
<enumeration value="GetLicensedCodecLines" />
<enumeration value="GetLicenseServerList" />

```



```

<enumeration value="GetNetParams" />
<enumeration value="GetNTPServer" />
<enumeration value="GetOMMCertificate" />
<enumeration value="GetOMPURL" />
<enumeration value="GetPARK" />
<enumeration value="GetPBXEnrolment" />
<enumeration value="GetPortRangeSIP" />
<enumeration value="GetPPDev" />
<enumeration value="GetPpProfile" />
<enumeration value="GetPPDevSummary" />
<enumeration value="GetPPFirmwareUpdate" />
<enumeration value="GetPPFirmwareUpdateOverview" />
<enumeration value="GetPPFirmwareUpdateStatus" />
<enumeration value="GetPPFirmwareURL" />
<enumeration value="GetPPLoginVariant" />
<enumeration value="GetPPState" />
<enumeration value="GetPPUser" />
<enumeration value="GetPPUserSummary" />
<enumeration value="GetPreserveUserDeviceRelation" />
<enumeration value="GetPublicKey" />
<enumeration value="GetReadyForConferencing" />
<enumeration value="GetRegistrationTrafficShaping" />
<enumeration value="GetRemoteAccess" />
<enumeration value="GetRemoteSystemDump" />
<enumeration value="GetRestrictedSubscriptionDuration" />
<enumeration value="GetRFP" />
<enumeration value="GetRFPCapture" />
<enumeration value="GetRFPCaptureList" />
<enumeration value="GetRFPipQuality" />
<enumeration value="GetRFFM" />
<enumeration value="GetRFFMediaStreamQuality" />
<enumeration value="GetRFPStatistic" />
<enumeration value="GetRFPStatisticConfig" />
<enumeration value="GetRFPsSummary" />
<enumeration value="GetRFPsSync" />
<enumeration value="GetRFPsSyncQuality" />
<enumeration value="GetRTP" />
<enumeration value="GetSARI" />
<enumeration value="GetSecureSIP" />
<enumeration value="GetSecureOMMCertificate" />
<enumeration value="GetSecureOMMCertificateServerImport" />
<enumeration value="GetSecurePROVCertificateServerImport" />
<enumeration value="GetSecureSIPCertificate" />
<enumeration value="GetSecureSIPCertificateServerImport" />
<enumeration value="GetSite" />
<enumeration value="GetSiteSummary" />
<enumeration value="GetSNMP" />
<enumeration value="GetSoftwareImageURL" />
<enumeration value="GetSpecialBranding" />
<enumeration value="GetStbState" />
<enumeration value="GetSuplServ" />
<enumeration value="GetSyslogServer" />
<enumeration value="GetSysStatisticConfig" />
<enumeration value="GetSysStatisticMinMax" />
<enumeration value="GetSysStatisticMinMaxRecord" />
<enumeration value="GetSysStatisticMinMaxSummary" />
<enumeration value="GetSysStatisticOccurrence" />
<enumeration value="GetSystemCredentials" />
<enumeration value="GetSystemName" />
<enumeration value="GetSystemProvUpdTrig" />
<enumeration value="GetSysToneScheme" />
<enumeration value="GetSysVoiceboxNum" />

```

```

<enumeration value="GetTimeZone"/>
<enumeration value="GetTimeZoneDetails"/>
<enumeration value="GetTimeZoneList"/>
<enumeration value="GetUsedConfigURL"/>
<enumeration value="GetUserDataServer"/>
<enumeration value="GetUserDeviceSyncOMM"/>
<enumeration value="GetUserMonitoring"/>
<enumeration value="GetWLANClients"/>
<enumeration value="GetWLANProfile"/>
<enumeration value="GetWLANRegDomain"/>
<enumeration value="GetWLANRegDomainList"/>
<enumeration value="GetVersions"/>
<enumeration value="GetVideoDev"/>
<enumeration value="GetVideoDevLink"/>
<enumeration value="GetVideoDevSummary"/>
<enumeration value="GetXMLApplication"/>
<enumeration value="Limits"/>
<enumeration value="ManualDBBackup"/>
<enumeration value="ManualUserDataImport"/>
<enumeration value="ManualDBRestore"/>
<enumeration value="Open"/>
<enumeration value="PARKFromServer"/>
<enumeration value="Ping"/>
<enumeration value="PutFile"/>
<enumeration value="RequestPositionInfo"/>
<enumeration value="ResetRFPMediaStreamQuality"/>
<enumeration value="ResetRFPStatistic"/>
<enumeration value="ResetRFPMediaStreamQuality"/>
<enumeration value="ResetSysStatistic"/>
<enumeration value="ResetTimeZoneDetails"/>
<enumeration value="SendMessage"/>
<enumeration value="SetAccount"/>
<enumeration value="SetACLEntry"/>
<enumeration value="SetAdditionalSettings"/>
<enumeration value="SetAdvancedSIP"/>
<enumeration value="SetAlarmTrigger"/>
<enumeration value="SetAutoDBBackup"/>
<enumeration value="SetBackupSIP"/>
<enumeration value="SetBasicSIP"/>
<enumeration value="SetBluetoothBeacon"/>
<enumeration value="SetBluetoothGlobalSettings"/>
<enumeration value="SetBluetoothSensitivity"/>
<enumeration value="SetConferenceRoom"/>
<enumeration value="SetConferenceServerSIP"/>
<enumeration value="SetConfigURL"/>
<enumeration value="SetCoreDumpURL"/>
<enumeration value="SetDECTAuthCode"/>
<enumeration value="SetDECTEncryption"/>
<enumeration value="SetDECTPagingAreaSize"/>
<enumeration value="SetDECTPowerLimit"/>
<enumeration value="SetDECTPpSettings"/>
<enumeration value="SetDECTRegDomain"/>
<enumeration value="SetDECTSubscriptionMode"/>
<enumeration value="SetDevAutoCreate"/>
<enumeration value="SetDigitTreatment"/>
<enumeration value="SetDTMF"/>
<enumeration value="SetEULAConfirm"/>
<enumeration value="SetFAC"/>
<enumeration value="SetFACList"/>
<enumeration value="SetFACPrefix"/>
<enumeration value="SetIMA"/>
<enumeration value="SetIntercomCallHandlingSIP"/>

```

```

<enumeration value="SetLDAP"/>
<enumeration value="SetLicenseServerList"/>
<enumeration value="SetNetParams"/>
<enumeration value="SetOMMCertificate"/>
<enumeration value="SetOMPURL"/>
<enumeration value="SetNTPServer"/>
<enumeration value="SetPARK"/>
<enumeration value="SetPBXEnrolment"/>
<enumeration value="SetPortRangeSIP"/>
<enumeration value="SetPPDev"/>
<enumeration value="SetPpProfile"/>
<enumeration value="SetPPFirmwareUpdate"/>
<enumeration value="SetPPFirmwareURL"/>
<enumeration value="SetPPLoginVariant"/>
<enumeration value="SetPPUser"/>
<enumeration value="SetPPUserDevRelation"/>
<enumeration value="SetPPUserTracking"/>
<enumeration value="SetPreserveUserDeviceRelation"/>
<enumeration value="SetRegistrationTrafficShaping"/>
<enumeration value="SetRemoteAccess"/>
<enumeration value="SetRemoteSystemDump"/>
<enumeration value="SetRestrictedSubscriptionDuration"/>
<enumeration value="SetRFP"/>
<enumeration value="SetRFPCapture"/>
<enumeration value="SetRFBPM"/>
<enumeration value="SetRTP"/>
<enumeration value="SetRTPConferenceStreamChg"/>
<enumeration value="SetSARI"/>
<enumeration value="SetSecureSIP"/>
<enumeration value="SetSecureOMMCertificate"/>
<enumeration value="SetSecureOMMCertificateServerImport"/>
<enumeration value="SetSecurePROVCertificateServerImport"/>
<enumeration value="SetSecureSIPCertificate"/>
<enumeration value="SetSecureSIPCertificateServerImport"/>
<enumeration value="SetSite"/>
<enumeration value="SetSNMP"/>
<enumeration value="SetSoftwareImageURL"/>
<enumeration value="SetSpecialBranding"/>
<enumeration value="SetSuplServ"/>
<enumeration value="SetSyslogServer"/>
<enumeration value="SetSystemCredentials"/>
<enumeration value="SetSystemName"/>
<enumeration value="SetSystemProvUpdTrig"/>
<enumeration value="SetSysToneScheme"/>
<enumeration value="SetSysVoiceboxNum"/>
<enumeration value="SetTimeZone"/>
<enumeration value="SetTimeZoneDetails"/>
<enumeration value="SetUserDataServer"/>
<enumeration value="SetUserDeviceSyncOMM"/>
<enumeration value="SetUserMonitoring"/>
<enumeration value="SetVideoDev"/>
<enumeration value="SetWLANProfile"/>
<enumeration value="SetWLANRegDomain"/>
<enumeration value="SetXMLApplication"/>
<enumeration value="SoftwareUpdate"/>
<enumeration value="Subscribe"/>
<enumeration value="SystemRestart"/>

</restriction>
</simpleType>

<complexType name="VersionType">

```

```

        <attribute name="element" type="tns:VersionElementType" use="required"/>
    </complexType>

    <complexType name="PublicKeyType">
        <attribute name="modulus" type="string" use="required"/>
        <attribute name="exponent" type="string" use="required"/>
    </complexType>

    <complexType name="GetRecordType">
        <attribute name="id" type="integer" use="required"/>
        <attribute name="maxRecords" type="integer" use="optional" default="0" />
    </complexType>

    <complexType name="GetSummaryRespType">
        <complexContent>
            <extension base="tns:RespType" />
            <attribute name="nRecords" type="integer" use="required"/>
            <attribute name="idFirst" type="integer" use="required"/>
        </complexContent>
    </complexType>

    <!-- ===== Basic related types ===== -->

    <complexType name="OpenType">
        <complexContent>
            <extension base="tns:ReqType">
                <attribute name="protocolVersion" type="integer" use="required"/>
                <attribute name="username" type="string" use="required"/>
                <attribute name="password" type="string" use="required"/>
            </extension>
        </complexContent>
    </complexType>

    <simpleType name="OmmStbStateType">
        <restriction base="string">
            <enumeration value="OK"/>
            <enumeration value="NotSynchronized"/>
            <enumeration value="DifferentOMMTypes"/>
            <enumeration value="DifferentOMMVersions"/>
            <enumeration value="None"/>
        </restriction>
    </simpleType>

    <complexType name="OpenRespType">
        <complexContent>
            <extension base="tns:RespType">
                <attribute name="protocolVersion" type="string" use="required"/>
                <attribute name="ommVersion" type="string" use="optional"/>
                <attribute name="axiClients" type="integer" use="optional"/>
                <attribute name="ommAxiSpecVersion" type="string" use="optional"/>
                <attribute name="minPPSwVersion1" type="string" use="optional"/>
                <attribute name="minPPSwVersion2" type="string" use="optional"/>
                <sequence>
                    <element name="accountState" type="tns:AccountStateType" use="optional"/>
                </sequence>
                <sequence>
                    <element name="permission" type="tns:PermissionType" use="optional"/>
                </sequence>
                <attribute name="ommStbState" type="tns:OmmStbStateType" use="required"/>
                <attribute name="ommStbAddr" type="string" use="optional"/>
                <attribute name="ommStream" type="string" use="optional"/>
                <attribute name="ommPlatform" type="string" use="optional"/>
            </extension>
        </complexContent>
    </complexType>

```

```

<attribute name="uptime" type="integer" use="optional"/>
<attribute name="EULAConfirm" type="boolean" use="optional"/>
<attribute name="publicKey" type="tns:PublicKeyType" use="required"/>
<attribute name="haveAdditionalSettings" type="boolean" use="optional"/>
<attribute name="haveAutoDB" type="boolean" use="optional"/>
<attribute name="haveBluetooth" type="boolean" use="optional"/>
<attribute name="haveConference" type="boolean" use="optional"/>
<attribute name="haveConfigPython" type="boolean" use="optional"/>
<attribute name="haveDECTRegDomain" type="boolean" use="optional"/>
<attribute name="haveDECTPpSettings" type="boolean" use="optional"/>
<attribute name="haveDigitTreatment" type="boolean" use="optional"/>
<attribute name="haveDigitTreatmentSimple" type="boolean" use="optional"/>
<attribute name="haveLDAP" type="boolean" use="optional"/>
<attribute name="havePPFirmwareUpdate" type="boolean" use="optional"/>
<attribute name="haveEnrolmentPBX" type="boolean" use="optional"/>
<attribute name="haveEnrolmentPP" type="boolean" use="optional"/>
<attribute name="haveEnrolmentRFP" type="boolean" use="optional"/>
<attribute name="haveExternalUserData" type="boolean" use="optional"/>
<attribute name="haveFACs" type="boolean" use="optional"/>
<attribute name="haveHCM" type="boolean" use="optional"/>
<attribute name="haveIMA" type="boolean" use="optional"/>
<attribute name="haveIPParamCnf" type="boolean" use="optional"/>
<attribute name="haveLicensing" type="boolean" use="optional"/>
<attribute name="haveLocating" type="boolean" use="optional"/>
<attribute name="haveNTP" type="boolean" use="optional"/>
<attribute name="haveOmmLogForward" type="boolean" use="optional"/>
<attribute name="haveOMP" type="boolean" use="optional"/>
<attribute name="havePagingAreas" type="boolean" use="optional"/>
<attribute name="havePARKFile" type="boolean" use="optional"/>
<attribute name="haveProtFTP" type="boolean" use="optional"/>
<attribute name="haveProtFTPS" type="boolean" use="optional"/>
<attribute name="haveProtHTTP" type="boolean" use="optional"/>
<attribute name="haveProtHTTPS" type="boolean" use="optional"/>
<attribute name="haveProtSCP" type="boolean" use="optional"/>
<attribute name="haveProtSFTP" type="boolean" use="optional"/>
<attribute name="haveProtTFTP" type="boolean" use="optional"/>
<attribute name="haveRFPpMCnf" type="boolean" use="optional"/>
<attribute name="haveRFPOMM" type="boolean" use="optional"/>
<attribute name="haveRTPPortBaseCnf" type="boolean" use="optional"/>
<attribute name="haveSARI" type="boolean" use="optional"/>
<attribute name="haveSemiAttendedTransfer" type="boolean" use="optional"/>
<attribute name="haveSIP" type="boolean" use="optional"/>
<attribute name="haveSIPSecure" type="boolean" use="optional"/>
<attribute name="haveSNMP" type="boolean" use="optional"/>
<attribute name="haveSpecialBranding" type="boolean" use="optional"/>
<attribute name="haveSRTP" type="boolean" use="optional"/>
<attribute name="haveSoftwareImageURL" type="boolean" use="optional"/>
<attribute name="haveSOSNumberPP" type="boolean" use="optional"/>
<attribute name="haveTZones" type="boolean" use="optional"/>
<attribute name="haveUnboundDevices" type="boolean" use="optional"/>
<attribute name="haveUMO" type="boolean" use="optional"/>
<attribute name="haveUserDeviceSync" type="boolean" use="optional"/>
<attribute name="haveVideo" type="boolean" use="optional"/>
<attribute name="haveVLAN" type="boolean" use="optional"/>
<attribute name="haveVoiceboxNumber" type="boolean" use="optional"/>
<attribute name="haveWLAN" type="boolean" use="optional"/>
<attribute name="haveXML" type="boolean" use="optional"/>
<attribute name="haveXMLDynamic" type="boolean" use="optional"/>
<attribute name="haveXMLCorpDir" type="boolean" use="optional"/>
</extension>
</complexContent>
</complexType>

```

```

<complexType name="VersionsType">
  <complexContent>
    <extension base="tns:ReqType">
      <sequence use="optional" >
        <attribute name="version" type="tns:VersionType" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="VersionsRespType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="ActivateRemoteSystemDump" type="string" use="optional"/>
      <attribute name="AddUserToConference" type="string" use="optional"/>
      <attribute name="CancelMessage" type="string" use="optional"/>
      <attribute name="ChangeUserInConference" type="string" use="optional"/>
      <attribute name="ConferenceConfirmation" type="string" use="optional"/>
      <attribute name="CreateAccount" type="string" use="optional"/>
      <attribute name="CreateACLEntry" type="string" use="optional"/>
      <attribute name="CreateAlarmTrigger" type="string" use="optional"/>
      <attribute name="CreateBluetoothBeacon" type="string" use="optional"/>
      <attribute name="CreateConference" type="string" use="optional"/>
      <attribute name="CreateConferenceRoom" type="string" use="optional"/>
      <attribute name="CreateDigitTreatment" type="string" use="optional"/>
      <attribute name="CreateFixedPP" type="string" use="optional"/>
      <attribute name="CreateLDAP" type="string" use="optional"/>
      <attribute name="CreatePPDev" type="string" use="optional"/>
      <attribute name="CreatePPUser" type="string" use="optional"/>
      <attribute name="CreateRFP" type="string" use="optional"/>
      <attribute name="CreateSite" type="string" use="optional"/>
      <attribute name="CreateVideoDev" type="string" use="optional"/>
      <attribute name="CreateWLANProfile" type="string" use="optional"/>
      <attribute name="CreateXMLApplication" type="string" use="optional"/>
      <attribute name="DBBackupToUSB" type="string" use="optional"/>
      <attribute name="DBRestoreFromUSB" type="string" use="optional"/>
      <attribute name="DeleteACLEntry" type="string" use="optional"/>
      <attribute name="DeleteAlarmTrigger" type="string" use="optional"/>
      <attribute name="DeleteBluetoothBeacon" type="string" use="optional"/>
      <attribute name="DeleteConference" type="string" use="optional"/>
      <attribute name="DeleteConferenceRoom" type="string" use="optional"/>
      <attribute name="DeleteDigitTreatment" type="string" use="optional"/>
      <attribute name="DeleteEventLogBuffer" type="string" use="optional"/>
      <attribute name="DeleteLDAP" type="string" use="optional"/>
      <attribute name="DeleteMessage" type="string" use="optional"/>
      <attribute name="DeletePPDev" type="string" use="optional"/>
      <attribute name="DeletePPUser" type="string" use="optional"/>
      <attribute name="DeleteRFP" type="string" use="optional"/>
      <attribute name="DeleteRFPCaptureList" type="string" use="optional"/>
      <attribute name="DeleteRFPCaptureListElem" type="string" use="optional"/>
      <attribute name="DeleteSite" type="string" use="optional"/>
      <attribute name="DeleteUserFromConference" type="string" use="optional"/>
      <attribute name="DeleteVideoDev" type="string" use="optional"/>
      <attribute name="DeleteWLANProfile" type="string" use="optional"/>
      <attribute name="DeleteXMLApplication" type="string" use="optional"/>
      <attribute name="EventAccountCnf" type="string" use="optional"/>
      <attribute name="EventAccountSummary" type="string" use="optional"/>
      <attribute name="EventACLCnf" type="string" use="optional"/>
      <attribute name="EventAdditionalSettingsCnf" type="string" use="optional"/>
      <attribute name="EventAdvancedSIPCnf" type="string" use="optional"/>
      <attribute name="EventAddUserToConference" type="string" use="optional"/>
    </extension>
  </complexContent>
</complexType>

```

```

<attribute name="EventAlarmCallProgress" type="string" use="optional"/>
<attribute name="EventAlarmTrigger" type="string" use="optional"/>
<attribute name="EventAlarmTriggerCnf" type="string" use="optional"/>
<attribute name="EventAutoDBBackupCnf" type="string" use="optional"/>
<attribute name="EventAutoDBBackupFileNameCnf" type="string" use="optional"/>
<attribute name="EventBackupSIPCnf" type="string" use="optional"/>
<attribute name="EventBasicSIPCnf" type="string" use="optional"/>
<attribute name="EventBluetoothClientStatistic" type="string" use="optional"/>
<attribute name="EventBluetoothBeaconCnf" type="string" use="optional"/>
<attribute name="EventBluetoothBeaconSummary" type="string" use="optional"/>
<attribute name="EventBluetoothGlobalSettingsCnf" type="string" use="optional"/>
<attribute name="EventBluetoothSensitivityCnf" type="string" use="optional"/>
<attribute name="EventChangeUserInConference" type="string" use="optional"/>
<attribute name="EventConferenceRoomCnf" type="string" use="optional"/>
<attribute name="EventConference" type="string" use="optional"/>
<attribute name="EventConferenceRelease" type="string" use="optional"/>
<attribute name="EventConferenceRequest" type="string" use="optional"/>
<attribute name="EventConferenceServerSIPCnf" type="string" use="optional"/>
<attribute name="EventConfigURLCnf" type="string" use="optional"/>
<attribute name="EventCoreDumpURLCnf" type="string" use="optional"/>
<attribute name="EventCreateConference" type="string" use="optional"/>
<attribute name="EventDbTransferState" type="string" use="optional"/>
<attribute name="EventDECTAuthCodeCnf" type="string" use="optional"/>
<attribute name="EventDECTEncryptionCnf" type="string" use="optional"/>
<attribute name="EventDECTPagingAreaSizeCnf" type="string" use="optional"/>
<attribute name="EventDECTPowerLimitCnf" type="string" use="optional"/>
<attribute name="EventDECTPpSettingsCnf" type="string" use="optional"/>
<attribute name="EventDECTRegDomainCnf" type="string" use="optional"/>
<attribute name="EventDECTSubscriptionMode" type="string" use="optional"/>
<attribute name="EventDeleteConference" type="string" use="optional"/>
<attribute name="EventDeleteUserFromConference" type="string" use="optional"/>
<attribute name="EventDevAutoCreateCnf" type="string" use="optional"/>
<attribute name="EventDigitTreatmentCnf" type="string" use="optional"/>
<attribute name="EventDTMFCnf" type="string" use="optional"/>
<attribute name="EventEULAConfirmCnf" type="string" use="optional"/>
<attribute name="EventEventLogEntry" type="string" use="optional"/>
<attribute name="EventFACCnf" type="string" use="optional"/>
<attribute name="EventFACPrefixCnf" type="string" use="optional"/>
<attribute name="EventFreeConferenceChannels" type="string" use="optional"/>
<attribute name="EventHealthState" type="string" use="optional"/>
<attribute name="EventIMACnf" type="string" use="optional"/>
<attribute name="EventIntercomCallHandlingSIPCnf" type="string" use="optional"/>
<attribute name="EventLDAPCnf" type="string" use="optional"/>
<attribute name="EventLicenseCnf" type="string" use="optional"/>
<attribute name="EventLicensedCodecLines" type="string" use="optional"/>
<attribute name="EventLicenseFile" type="string" use="optional"/>
<attribute name="EventLicenseServerListCnf" type="string" use="optional"/>
<attribute name="EventMessageConfirmation" type="string" use="optional"/>
<attribute name="EventMessageProgress" type="string" use="optional"/>
<attribute name="EventMessageQueueEmpty" type="string" use="optional"/>
<attribute name="EventMessageSend" type="string" use="optional"/>
<attribute name="EventNetParamsCnf" type="string" use="optional"/>
<attribute name="EventNTPServerCnf" type="string" use="optional"/>
<attribute name="EventOMMCertificateCnf" type="string" use="optional"/>
<attribute name="EventOMPURLCnf" type="string" use="optional"/>
<attribute name="EventPARKCnf" type="string" use="optional"/>
<attribute name="EventPARKFromServerResult" type="string" use="optional"/>
<attribute name="EventPBXEnrolmentCnf" type="string" use="optional"/>
<attribute name="EventPermissionChange" type="string" use="optional"/>
<attribute name="EventPortRangeSIPCnf" type="string" use="optional"/>
<attribute name="EventPositionHistory" type="string" use="optional"/>
<attribute name="EventPositionInfo" type="string" use="optional"/>

```

```

<attribute name="EventPositionRequest" type="string" use="optional"/>
<attribute name="EventPositionTrack" type="string" use="optional"/>
<attribute name="EventPPDevCnf" type="string" use="optional"/>
<attribute name="EventPpProfileCnf" type="string" use="optional"/>
<attribute name="EventPPFirmwareUpdateCnf" type="string" use="optional"/>
<attribute name="EventPPFirmwareUpdateOverview" type="string" use="optional"/>
<attribute name="EventPPFirmwareUpdateStatus" type="string" use="optional"/>
<attribute name="EventPPFirmwareURLCnf" type="string" use="optional"/>
<attribute name="EventPPLoginVariantCnf" type="string" use="optional"/>
<attribute name="EventPPState" type="string" use="optional"/>
<attribute name="EventPPTransaction" type="string" use="optional"/>
<attribute name="EventPPUserCnf" type="string" use="optional"/>
<attribute name="EventPPUserSummary" type="string" use="optional"/>
<attribute name="EventPreserveUserDeviceRelationCnf" type="string" use="optional"/>
<attribute name="EventReadyForConferencing" type="string" use="optional"/>
<attribute name="EventRegistrationTrafficShapingCnf" type="string" use="optional"/>
<attribute name="EventRemoteAccessCnf" type="string" use="optional"/>
<attribute name="EventRemoteSystemDumpCnf" type="string" use="optional"/>
<attribute name="EventRestrictedSubscriptionDurationCnf" type="string" use="optional"/>
<attribute name="EventRFPConf" type="string" use="optional"/>
<attribute name="EventRFPCapture" type="string" use="optional"/>
<attribute name="EventRFPCaptureCnf" type="string" use="optional"/>
<attribute name="EventRFPConnectAttempt" type="string" use="optional"/>
<attribute name="EventRFPDetails" type="string" use="optional"/>
<attribute name="EventRFPipQuality" type="string" use="optional"/>
<attribute name="EventRFPMCnf" type="string" use="optional"/>
<attribute name="EventRFPMediaStreamQuality" type="string" use="optional"/>
<attribute name="EventRFPState" type="string" use="optional"/>
<attribute name="EventRFPSummary" type="string" use="optional"/>
<attribute name="EventRFPsSyncQuality" type="string" use="optional"/>
<attribute name="EventRFPsSyncRel" type="string" use="optional"/>
<attribute name="EventRTPConferenceStreamChg" type="string" use="optional"/>
<attribute name="EventRTPCnf" type="string" use="optional"/>
<attribute name="EventSARICnf" type="string" use="optional"/>
<attribute name="EventSecureOMMCertificateServerImportCnf" type="string" use="optional"/>
<attribute name="EventSecurePROVCertificateServerImportCnf" type="string" use="optional"/>
<attribute name="EventSecureSIPCertificateCnf" type="string" use="optional"/>
<attribute name="EventSecureSIPCertificateServerImportCnf" type="string" use="optional"/>
<attribute name="EventSecureSIPCnf" type="string" use="optional"/>
<attribute name="EventSiteCnf" type="string" use="optional"/>
<attribute name="EventSiteSummary" type="string" use="optional"/>
<attribute name="EventSNMPCnf" type="string" use="optional"/>
<attribute name="EventSoftwareImageURLCnf" type="string" use="optional"/>
<attribute name="EventSpecialBrandingCnf" type="string" use="optional"/>
<attribute name="EventStbStateChange" type="string" use="optional"/>
<attribute name="EventSuplServCnf" type="string" use="optional"/>
<attribute name="EventSyslogServerCnf" type="string" use="optional"/>
<attribute name="EventSystemCredentialsCnf" type="string" use="optional"/>
<attribute name="EventSystemNameCnf" type="string" use="optional"/>
<attribute name="EventSystemProvUpdTrigCnf" type="string" use="optional"/>
<attribute name="EventSysToneSchemeCnf" type="string" use="optional"/>
<attribute name="EventSysVoiceboxNumCnf" type="string" use="optional"/>
<attribute name="EventTimeZoneCnf" type="string" use="optional"/>
<attribute name="EventTimeZoneDetailsCnf" type="string" use="optional"/>
<attribute name="EventTimeZoneList" type="string" use="optional"/>
<attribute name="EventUsedConfigURL" type="string" use="optional"/>
<attribute name="EventUserDataServerCnf" type="string" use="optional"/>
<attribute name="EventUserDeviceSyncOMMCnf" type="string" use="optional"/>
<attribute name="EventUserMonitoringCnf" type="string" use="optional"/>
<attribute name="EventVideoDevCnf" type="string" use="optional"/>
<attribute name="EventVideoDevLink" type="string" use="optional"/>
<attribute name="EventVideoDevSummary" type="string" use="optional"/>

```



```

<attribute name="EventWLANClient" type="string" use="optional"/>
<attribute name="EventWLANProfileCnf" type="string" use="optional"/>
<attribute name="EventWLANRegDomainCnf" type="string" use="optional"/>
<attribute name="EventXMLApplicationCnf" type="string" use="optional"/>
<attribute name="GenerateHealthStateAlarmTriggers" type="string" use="optional"/>
<attribute name="GetAccount" type="string" use="optional"/>
<attribute name="GetAccountSummary" type="string" use="optional"/>
<attribute name="GetACLEntry" type="string" use="optional"/>
<attribute name="GetActivePPDev" type="string" use="optional"/>
<attribute name="GetAdditionalSettings" type="string" use="optional"/>
<attribute name="GetAdvancedSIP" type="string" use="optional"/>
<attribute name="GetAlarmTrigger" type="string" use="optional"/>
<attribute name="GetAlarmTriggerSummary" type="string" use="optional"/>
<attribute name="GetAutoDBBackup" type="string" use="optional"/>
<attribute name="GetAutoDBBackupFileName" type="string" use="optional"/>
<attribute name="GetBackupSIP" type="string" use="optional"/>
<attribute name="GetBasicSIP" type="string" use="optional"/>
<attribute name="GetBluetoothBeacon" type="string" use="optional"/>
<attribute name="GetBluetoothBeaconSummary" type="string" use="optional"/>
<attribute name="GetBluetoothGlobalSettings" type="string" use="optional"/>
<attribute name="GetBluetoothSensitivity" type="string" use="optional"/>
<attribute name="GetConferenceRoom" type="string" use="optional"/>
<attribute name="GetConferenceServerSIP" type="string" use="optional"/>
<attribute name="GetConfigURL" type="string" use="optional"/>
<attribute name="GetCoreDumpURL" type="string" use="optional"/>
<attribute name="GetDbTransferState" type="string" use="optional"/>
<attribute name="GetDECTAuthCode" type="string" use="optional"/>
<attribute name="GetDECTEncryption" type="string" use="optional"/>
<attribute name="GetDECTPagingAreaSize" type="string" use="optional"/>
<attribute name="GetDECTPowerLimit" type="string" use="optional"/>
<attribute name="GetDECTPpSettings" type="string" use="optional"/>
<attribute name="GetDECTRegDomain" type="string" use="optional"/>
<attribute name="GetDECTSubscriptionMode" type="string" use="optional"/>
<attribute name="GetDevAutoCreate" type="string" use="optional"/>
<attribute name="GetDigitTreatment" type="string" use="optional"/>
<attribute name="GetDigitTreatmentSummary" type="string" use="optional"/>
<attribute name="GetDTMF" type="string" use="optional"/>
<attribute name="GetEULAConfirm" type="string" use="optional"/>
<attribute name="GetEventLogBuffer" type="string" use="optional"/>
<attribute name="GetFACList" type="string" use="optional"/>
<attribute name="GetFACPrefix" type="string" use="optional"/>
<attribute name="GetFile" type="string" use="optional"/>
<attribute name="GetFlashMemUsage" type="string" use="optional"/>
<attribute name="GetFreeConferenceChannels" type="string" use="optional"/>
<attribute name="GetG729ChannelsForConference" type="string" use="optional"/>
<attribute name="GetHealthState" type="string" use="optional"/>
<attribute name="GetIMA" type="string" use="optional"/>
<attribute name="GetIntercomCallHandlingSIP" type="string" use="optional"/>
<attribute name="GetLastPPDevAction" type="string" use="optional"/>
<attribute name="GetLDAP" type="string" use="optional"/>
<attribute name="GetLicense" type="string" use="optional"/>
<attribute name="GetLicensedCodeclines" type="string" use="optional"/>
<attribute name="GetLicenseServerList" type="string" use="optional"/>
<attribute name="GetNetParams" type="string" use="optional"/>
<attribute name="GetNTPServer" type="string" use="optional"/>
<attribute name="GetOMMCertificate" type="string" use="optional"/>
<attribute name="GetOMPURL" type="string" use="optional"/>
<attribute name="GetPARK" type="string" use="optional"/>
<attribute name="GetPBXEnrolment" type="string" use="optional"/>
<attribute name="GetPortRangeSIP" type="string" use="optional"/>
<attribute name="GetPPDev" type="string" use="optional"/>
<attribute name="GetPpProfile" type="string" use="optional"/>

```

```

<attribute name="GetPPDevSummary" type="string" use="optional"/>
<attribute name="GetPPFirmwareUpdate" type="string" use="optional"/>
<attribute name="GetPPFirmwareUpdateOverview" type="string" use="optional"/>
<attribute name="GetPPFirmwareUpdateStatus" type="string" use="optional"/>
<attribute name="GetPPFirmwareURL" type="string" use="optional"/>
<attribute name="GetPPLoginVariant" type="string" use="optional"/>
<attribute name="GetPPState" type="string" use="optional"/>
<attribute name="GetPPUser" type="string" use="optional"/>
<attribute name="GetPPUserSummary" type="string" use="optional"/>
<attribute name="GetPreserveUserDeviceRelation" type="string"/>
<attribute name="GetPublicKey" type="string" use="optional"/>
<attribute name="GetReadyForConferencing" type="string" use="optional"/>
<attribute name="GetRegistrationTrafficShaping" type="string" use="optional"/>
<attribute name="GetRemoteAccess" type="string" use="optional"/>
<attribute name="GetRemoteSystemDump" type="string" use="optional"/>
<attribute name="GetRestrictedSubscriptionDuration" type="string" use="optional"/>
<attribute name="GetRFP" type="string" use="optional"/>
<attribute name="GetRFPCapture" type="string" use="optional"/>
<attribute name="GetRFPCaptureList" type="string" use="optional"/>
<attribute name="GetRFPM" type="string" use="optional"/>
<attribute name="GetRFPStatistic" type="string" use="optional"/>
<attribute name="GetRFPStatisticConfig" type="string" use="optional"/>
<attribute name="GetRFPSummary" type="string" use="optional"/>
<attribute name="GetRFPSync" type="string" use="optional"/>
<attribute name="GetRTP" type="string" use="optional"/>
<attribute name="GetSARI" type="string" use="optional"/>
<attribute name="GetSecureSIP" type="string" use="optional"/>
<attribute name="GetSecureOMMCertificateServerImport" type="string" use="optional"/>
<attribute name="GetSecurePROVCertificateServerImport" type="string" use="optional"/>
<attribute name="GetSecureSIPCertificate" type="string" use="optional"/>
<attribute name="GetSecureSIPCertificateServerImport" type="string" use="optional"/>
<attribute name="GetSite" type="string" use="optional"/>
<attribute name="GetSiteSummary" type="string" use="optional"/>
<attribute name="GetSNMP" type="string" use="optional"/>
<attribute name="GetSoftwareImageURL" type="string" use="optional"/>
<attribute name="GetSpecialBranding" type="string" use="optional"/>
<attribute name="GetStbState" type="string" use="optional"/>
<attribute name="GetSuplServ" type="string" use="optional"/>
<attribute name="GetSyslogServer" type="string" use="optional"/>
<attribute name="GetSysStatisticConfig" type="string" use="optional"/>
<attribute name="GetSysStatisticMinMax" type="string" use="optional"/>
<attribute name="GetSysStatisticMinMaxRecord" type="string" use="optional"/>
<attribute name="GetSysStatisticMinMaxSummary" type="string" use="optional"/>
<attribute name="GetSysStatisticOccurrence" type="string" use="optional"/>
<attribute name="GetSystemCredentials" type="string" use="optional"/>
<attribute name="GetSystemName" type="string" use="optional"/>
<attribute name="GetSystemProvUpdTrig" type="string" use="optional"/>
<attribute name="GetSysToneScheme" type="string" use="optional"/>
<attribute name="GetSysVoiceboxNum" type="string" use="optional"/>
<attribute name="GetTimeZone" type="string" use="optional"/>
<attribute name="GetTimeZoneDetails" type="string" use="optional"/>
<attribute name="GetTimeZoneList" type="string" use="optional"/>
<attribute name="GetUsedConfigURL" type="string" use="optional"/>
<attribute name="GetUserDataServer" type="string" use="optional"/>
<attribute name="GetUserDeviceSyncOMM" type="string" use="optional"/>
<attribute name="GetUserMonitoring" type="string" use="optional"/>
<attribute name="GetVersions" type="string" use="optional"/>
<attribute name="GetVideoDev" type="string" use="optional"/>
<attribute name="GetVideoDevLink" type="string" use="optional"/>
<attribute name="GetVideoDevSummary" type="string" use="optional"/>
<attribute name="GetWLANclients" type="string" use="optional"/>
<attribute name="GetWLANProfile" type="string" use="optional"/>

```

```

<attribute name="GetWLANRegDomain" type="string" use="optional"/>
<attribute name="GetWLANRegDomainList" type="string" use="optional"/>
<attribute name="GetXMLApplication" type="string" use="optional"/>
<attribute name="Limits" type="string" use="optional"/>
<attribute name="ManualDBBackup" type="string" use="optional"/>
<attribute name="ManualUserDataImport" type="string" use="optional"/>
<attribute name="ManualDBRestore" type="string" use="optional"/>
<attribute name="Open" type="string" use="optional"/>
<attribute name="PARKFromServer" type="string" use="optional"/>
<attribute name="Ping" type="string" use="optional"/>
<attribute name="PutFile" type="string" use="optional"/>
<attribute name="RequestPositionInfo" type="string" use="optional"/>
<attribute name="ResetRFPStatistic" type="string" use="optional"/>
<attribute name="ResetSysStatistic" type="string" use="optional"/>
<attribute name="ResetTimeZoneDetails" type="string" use="optional"/>
<attribute name="SendMessage" type="string" use="optional"/>
<attribute name="SetAccount" type="string" use="optional"/>
<attribute name="SetACLEntry" type="string" use="optional"/>
<attribute name="SetAdditionalSettings" type="string" use="optional"/>
<attribute name="SetAdvancedSIP" type="string" use="optional"/>
<attribute name="SetAlarmTrigger" type="string" use="optional"/>
<attribute name="SetAutoDBBackup" type="string" use="optional"/>
<attribute name="SetBackupSIP" type="string" use="optional"/>
<attribute name="SetBasicSIP" type="string" use="optional"/>
<attribute name="SetBluetoothBeacon" type="string" use="optional"/>
<attribute name="SetBluetoothGlobalSettings" type="string" use="optional"/>
<attribute name="SetBluetoothSensitivity" type="string" use="optional"/>
<attribute name="SetConferenceRoom" type="string" use="optional"/>
<attribute name="SetConferenceServerSIP" type="string" use="optional"/>
<attribute name="SetConfigURL" type="string" use="optional"/>
<attribute name="SetCoreDumpURL" type="string" use="optional"/>
<attribute name="SetDECTAuthCode" type="string" use="optional"/>
<attribute name="SetDECTEncryption" type="string" use="optional"/>
<attribute name="SetDECTPagingAreaSize" type="string" use="optional"/>
<attribute name="SetDECTPowerLimit" type="string" use="optional"/>
<attribute name="SetDECTPpSettings" type="string" use="optional"/>
<attribute name="SetDECTRegDomain" type="string" use="optional"/>
<attribute name="SetDECTSubscriptionMode" type="string" use="optional"/>
<attribute name="SetDevAutoCreate" type="string" use="optional"/>
<attribute name="SetDigitTreatment" type="string" use="optional"/>
<attribute name="SetDTMF" type="string" use="optional"/>
<attribute name="SetEULAConfirm" type="string" use="optional"/>
<attribute name="SetFAC" type="string" use="optional"/>
<attribute name="SetFACList" type="string" use="optional"/>
<attribute name="SetFACPrefix" type="string" use="optional"/>
<attribute name="SetIMA" type="string" use="optional"/>
<attribute name="SetIntercomCallHandlingSIP" type="string" use="optional"/>
<attribute name="SetLDAP" type="string" use="optional"/>
<attribute name="SetLicenseServerList" type="string" use="optional"/>
<attribute name="SetNetParams" type="string" use="optional"/>
<attribute name="SetNTPServer" type="string" use="optional"/>
<attribute name="SetOMMCertificate" type="string" use="optional"/>
<attribute name="SetOMPURL" type="string" use="optional"/>
<attribute name="SetPARK" type="string" use="optional"/>
<attribute name="SetPBXEnrolment" type="string" use="optional"/>
<attribute name="SetPortRangeSIP" type="string" use="optional"/>
<attribute name="SetPPDev" type="string" use="optional"/>
<attribute name="SetPpProfile" type="string" use="optional"/>
<attribute name="SetPPFirmwareUpdate" type="string" use="optional"/>
<attribute name="SetPPFirmwareURL" type="string" use="optional"/>
<attribute name="SetPPLoginVariant" type="string" use="optional"/>
<attribute name="SetPPUser" type="string" use="optional"/>

```

```

    <attribute name="SetPPUserDevRelation" type="string" use="optional"/>
    <attribute name="SetPPUserTracking" type="string" use="optional"/>
    <attribute name="SetPreserveUserDeviceRelation" type="string"/>
    <attribute name="SetRegistrationTrafficShaping" type="string" use="optional"/>
    <attribute name="SetRemoteAccess" type="string" use="optional"/>
    <attribute name="SetRemoteSystemDump" type="string" use="optional"/>
    <attribute name="SetRestrictedSubscriptionDuration" type="string" use="optional"/>
    <attribute name="SetRFP" type="string" use="optional"/>
    <attribute name="SetRFPCapture" type="string" use="optional"/>
    <attribute name="SetRFPM" type="string" use="optional"/>
    <attribute name="SetRTP" type="string" use="optional"/>
    <attribute name="SetRTPConferenceStreamChg" type="string" use="optional"/>
    <attribute name="SetSARI" type="string" use="optional"/>
    <attribute name="SetSecureSIP" type="string" use="optional"/>
    <attribute name="SetSecureOMMCertificateServerImport" type="string" use="optional"/>
    <attribute name="SetSecurePROVCertificateServerImport" type="string" use="optional"/>
    <attribute name="SetSecureSIPCertificate" type="string" use="optional"/>
    <attribute name="SetSecureSIPCertificateServerImport" type="string" use="optional"/>
    <attribute name="SetSite" type="string" use="optional"/>
    <attribute name="SetSNMP" type="string" use="optional"/>
    <attribute name="SetSoftwareImageURL" type="string" use="optional"/>
    <attribute name="SetSpecialBranding" type="string" use="optional"/>
    <attribute name="SetSuplServ" type="string" use="optional"/>
    <attribute name="SetSyslogServer" type="string" use="optional"/>
    <attribute name="SetSystemCredentials" type="string" use="optional"/>
    <attribute name="SetSystemName" type="string" use="optional"/>
    <attribute name="SetSystemProvUpdTrig" type="string" use="optional"/>
    <attribute name="SetSysToneScheme" type="string" use="optional"/>
    <attribute name="SetSysVoiceboxNum" type="string" use="optional"/>
    <attribute name="SetTimeZone" type="string" use="optional"/>
    <attribute name="SetTimeZoneDetails" type="string" use="optional"/>
    <attribute name="SetUserDataServer" type="string" use="optional"/>
    <attribute name="SetUserDeviceSyncOMM" type="string" use="optional"/>
    <attribute name="SetUserMonitoring" type="string" use="optional"/>
    <attribute name="SetVideoDev" type="string" use="optional"/>
    <attribute name="SetWLANProfile" type="string" use="optional"/>
    <attribute name="SetWLANRegDomain" type="string" use="optional"/>
    <attribute name="SetXMLApplication" type="string" use="optional"/>
    <attribute name="SoftwareUpate" type="string" use="optional"/>
    <attribute name="Subscribe" type="string" use="optional"/>
    <attribute name="SystemRestart" type="string" use="optional"/>

    </extension>
  </complexContent>
</complexType>

<complexType name="LimitsRespType">
  <attribute name="axiClients" type="integer"/>
  <attribute name="codec" type="integer"/>
  <attribute name="ldapServer" type="integer"/>
  <attribute name="licLatency" type="integer"/>
  <attribute name="ppnNum" type="integer"/>
  <attribute name="certificateNum" type="integer"/>
  <attribute name="records" type="integer"/>
  <attribute name="rfpNum" type="integer"/>
  <attribute name="dnsServer" type="integer"/>
  <attribute name="ntpServer" type="integer"/>
  <attribute name="site" type="integer"/>
  <attribute name="ppVideoDevNum" type="integer"/>
  <attribute name="ppProfileNum" type="integer"/>
  <attribute name="ssidWlanKey" type="integer"/>
  <attribute name="trigger" type="integer"/>

```

```

<attribute name="urlLen" type="integer"/>
<attribute name="userId" type="integer"/>
<attribute name="wlanMacFilter" type="integer"/>
<attribute name="xmlBuiltInAppl" type="integer"/>
<attribute name="xmlDynAppl" type="integer"/>
<attribute name="xmlCorpDirAppl" type="integer"/>
<attribute name="wlanClients" type="integer"/>
<attribute name="wlanProfiles" type="integer"/>
<attribute name="BluetoothBeacons" type="integer"/>
<attribute name="conferenceRooms" type="integer"/>
<attribute name="limitRTT1" type="integer"/>
<attribute name="limitRTT2" type="integer"/>
<attribute name="limitRTT3" type="integer"/>
<attribute name="limitRTT4" type="integer"/>
</complexType>

<simpleType name="CmdType">
  <restriction base="string">
    <enumeration value="On"/>
    <enumeration value="Off"/>
  </restriction>
</simpleType>

<simpleType name="EventType">
  <restriction base="string">
    <enumeration value="AccountCnf"/>
    <enumeration value="AccountSummary"/>
    <enumeration value="ACLCnf"/>
    <enumeration value="AlarmCallProgress"/>
    <enumeration value="AlarmTrigger"/>
    <enumeration value="AlarmTriggerCnf"/>
    <enumeration value="AutoDBCnf"/>
    <enumeration value="BluetoothCnf"/>
    <enumeration value="BluetoothBeaconSumCnf"/>
    <enumeration value="BluetoothBeaconSumState"/>
    <enumeration value="Conference"/>
    <enumeration value="ConferenceCnf"/>
    <enumeration value="DECTSubscriptionMode"/>
    <enumeration value="DigitTreatmentCnf"/>
    <enumeration value="EventLog"/>
    <enumeration value="FACnf"/>
    <enumeration value="LicensedCodecLines"/>
    <enumeration value="LicenseFile"/>
    <enumeration value="Locating"/>
    <enumeration value="LocatingBluetooth"/>
    <enumeration value="MessageConfirmation"/>
    <enumeration value="MessageProgress"/>
    <enumeration value="MessageQueueEmpty"/>
    <enumeration value="MessageSend"/>
    <enumeration value="ParkServerResult"/>
    <enumeration value="PositionTrack"/>
    <enumeration value="PPBtState"/>
    <enumeration value="PPDevCnf"/>
    <enumeration value="PPFirmwareUpdateCnf"/>
    <enumeration value="PPFirmwareUpdateOverview"/>
    <enumeration value="PPFirmwareUpdateState"/>
    <enumeration value="PPFirmwareState"/>
    <enumeration value="PPHCM"/>
    <enumeration value="PPState"/>
    <enumeration value="PPCallState"/>
    <enumeration value="PPDevSummary"/>
    <enumeration value="PPDevState"/>
  </restriction>
</simpleType>

```

```

<enumeration value="PPCallStateCalling"/>
<enumeration value="PPCallStatePaging"/>
<enumeration value="PPCallStateConnected"/>
<enumeration value="PPCallStateIdle"/>
<enumeration value="PPCallStateNone"/>
<enumeration value="PPCallStateRinging"/>
<enumeration value="PPDevStateOnHook"/>
<enumeration value="PPDevStateSilentCharging"/>
<enumeration value="PPDevStateBatteryLevel"/>
<enumeration value="PPDevStateSwVersion"/>
<enumeration value="PPSipState"/>
<enumeration value="PPSipStateRegEvent"/>
<enumeration value="PPSipStateEndEvent"/>
<enumeration value="PPSipStateNotifyEvent"/>
<enumeration value="PPSipStateRegistered"/>
<enumeration value="PPSipStateServerType"/>
<enumeration value="PPSipStateServerAddr"/>
<enumeration value="PPSipStateServerPort"/>
<enumeration value="PPTransaction"/>
<enumeration value="PPTransactionLinkEstablish"/>
<enumeration value="PPTransactionRelease"/>
<enumeration value="PPTransactionEstablish"/>
<enumeration value="PPTransactionPPNotFound"/>
<enumeration value="PPTransactionPagingStarted"/>
<enumeration value="PPTransactionReleaseFromPP"/>
<enumeration value="PPTransactionPPSetupRejected"/>
<enumeration value="PPTransactionComsRelease"/>
<enumeration value="PPTransactionComsEstablish"/>
<enumeration value="PPTransactionSsFac"/>
<enumeration value="PPTransactionSsRelease"/>
<enumeration value="PPTransactionSsEstablish"/>
<enumeration value="PPTransactionConnHandover"/>
<enumeration value="PPTransactionLocReg"/>
<enumeration value="PPTransactionDetach"/>
<enumeration value="PPUserCnf"/>
<enumeration value="PPUserSummary"/>
<enumeration value="RFP Cnf"/>
<enumeration value="RFPConnectAttempt"/>
<enumeration value="RFPDetails"/>
<enumeration value="RFPipQuality"/>
<enumeration value="RFPmsQuality"/>
<enumeration value="RFPState"/>
<enumeration value="RFPSummary"/>
<enumeration value="RFPsync"/>
<enumeration value="RFPsyncQuality"/>
<enumeration value="Roaming"/>
<enumeration value="SiteCnf"/>
<enumeration value="SiteSummary"/>
<enumeration value="SuplCallFeatures"/>
<enumeration value="SystemCnf"/>
<enumeration value="SystemState"/>
<enumeration value="UserDataImport"/>
<enumeration value="VideoDev"/>
<enumeration value="VideoDevSumCnf"/>
<enumeration value="VideoDevSumState"/>
<enumeration value="WLANClient"/>
<enumeration value="WLANProfileCnf"/>
</restriction>
</simpleType>

<complexType name="SubscribeType">
  <sequence minOccurs="1" maxOccurs="20">

```

```

        <element name="e" type="tns:SubscribeCmdType" />
    </sequence>
    <attribute name="seq" type="integer" use="optional" />
</complexType>

<complexType name="SubscribeCmdType">
    <attribute name="cmd" type="tns:CmdType" use="required" />
    <attribute name="eventType" type="tns:EventType" use="required" />
    <attribute name="ppn" type="integer" use="optional" />
    <attribute name="uid" type="integer" use="optional" />
    <attribute name="rfpId" type="integer" use="optional" />
    <attribute name="omm" type="integer" use="optional" />
    <attribute name="trigger" type="string" use="optional" />
    <attribute name="scheme" type="string" use="optional" />
</complexType>

<complexType name="SubscribeRespType">
    <complexContent>
        <extension base="tns:RespType">
            <attribute name="eventType" type="tns:EventType" use="optional" />
        </extension>
    </complexContent>
</complexType>

<complexType name="StbStateType">
    <attribute name="ommStbState" type="tns:OmmStbStateType" use="required" />
    <attribute name="ommStbAddr" type="string" use="optional" />
</complexType>

<complexType name="StbStateRespType">
    <complexContent>
        <extension base="tns:RespType" />
        <extension base="tns:StbStateType" use="required" />
    </complexContent>
</complexType>

<complexType name="PermissionChangeEvent">
    <complexContent>
        <extension base="tns:RespType">
            <attribute name="permission" type="tns:PermissionType" use="optional" />
        </extension>
    </complexContent>
</complexType>

<complexType name="GetPublicKeyRespType">
    <complexContent>
        <extension base="tns:RespType">
            <attribute name="modulus" type="string" use="required" />
            <attribute name="exponent" type="string" use="required" />
        </extension>
    </complexContent>
</complexType>

<simpleType name="conferenceServerType">
    <restriction base="string">
        <enumeration value="None" />
        <enumeration value="Integrated" />
        <enumeration value="External" />
        <enumeration value="ExternalBlindTransfer" />
        <enumeration value="Global" />
    </restriction>
</simpleType>

```

```

<!-- ===== license related types ===== -->

<complexType name="LicenseDataType">
  <attribute name="key" type="string" use="required"/>
  <attribute name="number" type="integer" use="required"/>
  <attribute name="validation" type="string" />
  <attribute name="systemLicenseVersion" type="string" use="required"/>
  <attribute name="systemLicenseSRTP" type="boolean" use="required"/>
  <attribute name="systemLicenseDECTSec" type="boolean" use="required"/>
  <attribute name="messagingLicenseRcvMsgs" type="boolean" use="required"/>
  <attribute name="locatingLicense" type="boolean" use="required"/>
</complexType>

<complexType name="LicenseRFPType">
  <enumeration value="id"/>
  <enumeration value="ethAddr"/>
  <enumeration value="connected"/>
</complexType>

<complexType name="LicenseTypeType">
  <enumeration value="small"/>
  <enumeration value="standard"/>
</complexType>

<complexType name="LicenseStateType">
  <enumeration value="noLicense"/>
  <enumeration value="inactiveLicense"/>
  <enumeration value="inactiveLicenseBlocked"/>
  <enumeration value="noRedundancyLicense"/>
  <enumeration value="noRedundancyLicenseBlocked"/>
  <enumeration value="activeLicense"/>
  <enumeration value="activeLicenseBlocked"/>
</complexType>

<complexType name="LicenseViolationType">
  <enumeration value="noViolation"/>
  <enumeration value="noLicense"/>
  <enumeration value="numRFPs"/>
  <enumeration value="swRelease"/>
  <enumeration value="numSendMsgs"/>
  <enumeration value="numLocatables"/>
</complexType>

<complexType name="LicenseType">
  <complexContent>
    <attribute name="type" type="tns:LicenseTypeType" use="optional"/>
    <attribute name="state" type="tns:LicenseStateType" use="optional"/>
    <sequence>
      <attribute name="violation" type="tns:LicenseViolationType" use="optional"/>
    </sequence>
    <attribute name="latency" type="integer" use="optional"/>
    <attribute name="park" type="string" use="optional"/>
    <sequence>
      <attribute name="licenseRfp" type="tns:LicenseRFPType" use="optional"/>
    </sequence>
    <attribute name="sysLicense" type="tns:LicenseDataType" use="optional"/>
    <attribute name="msgLicense" type="tns:LicenseDataType" use="optional"/>
    <attribute name="locLicense" type="tns:LicenseDataType" use="optional"/>
    <attribute name="G729License" type="tns:LicenseDataType" use="optional"/>
  </complexContent>

```



```

</complexType>

<complexType name="LicenseRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:LicenseType"> </extension>
  </complexContent>
</complexType>

<complexType name="LicensedCodecLinesType">
  <attribute name="numberOfLines" type="integer" use="required"/>
</complexType>

<complexType name="LicensedCodecLinesRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:LicensedCodecLinesType"> </extension>
  </complexContent>
</complexType>

<complexType name="LicensedCodecLinesRespType">
  <complexContent>
    <attribute name="server" type="string" use="optional"/>
    <attribute name="port" type="integer" use="optional"/>
    <attribute name="serverStandby" type="string" use="optional"/>
    <attribute name="portStandby" type="integer" use="optional"/>
  </complexContent>
</complexType>

<complexType name="LicenseServerListType">
  <complexContent>
    <attribute name="serverActive" type="boolean" use="optional"/>
    <attribute name="installationId" type="integer" use="optional"/>
    <attribute name="licenseServer1" type="tns:LicenseServerListElementType" use="optional"/>
    <attribute name="licenseServer2" type="tns:LicenseServerListElementType" use="optional"/>
    <attribute name="licenseServer3" type="tns:LicenseServerListElementType" use="optional"/>
  </complexContent>
</complexType>

<complexType name="SetLicenseServerListType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:LicenseServerListType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetLicenseServerListRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:LicenseServerListType"> </extension>
  </complexContent>
</complexType>

<complexType name="LicenseFileTypeType">
  <enumeration value="LicenseFile"/>
  <enumeration value="LicenseDataFile"/>
</complexType>

<complexType name="LicenseFileType">
  <complexContent>
    <attribute name="type" type="tns:LicenseFileTypeType"/>
  </complexContent>

```

```

</complexType>

<!-- ===== account related types ===== -->

<simpleType name="AccountAgingType">
  <restriction base="string">
    <enumeration value="time3Months"/>
    <enumeration value="time6Months"/>
    <enumeration value="count50Logins"/>
    <enumeration value="count100Logins"/>
    <enumeration value="none"/>
  </restriction>
</simpleType>

<simpleType name="AccountStateTypeType">
  <restriction base="string">
    <enumeration value="dontCare"/>
    <enumeration value="changed"/>
    <enumeration value="notChanged"/>
    <enumeration value="expired"/>
  </restriction>
</simpleType>

<complexType name="AccountType">
  <complexContent>
    <attribute name="id" type="integer" />
    <attribute name="username" type="string" />
    <attribute name="password" type="string" />
    <attribute name="oldPassword" type="string" />
    <sequence>
      <attribute name="permission" type="tns:PermissionType"/>
    </sequence>
    <attribute name="active" type="boolean" />
    <attribute name="aging" type="tns:AccountAgingType"/>
    <attribute name="expire" type="integer" />
    <attribute name="state" type="tns:AccountStateTypeType" />
  </complexContent>
</complexType>

<complexType name="CreateAccountType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:AccountType" />
  </complexContent>
</complexType>

<complexType name="DeleteAccountType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer" use="required"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetAccountType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:GetRecordType" />
  </complexContent>
</complexType>

<complexType name="SetAccountType">

```

```

    <complexContent>
      <extension base="tns:ReqType"> </extension>
      <extension base="tns:AccountType"> </extension>
    </complexContent>
  </complexType>

  <complexType name="CreateGetSetAccountRespType">
    <complexContent>
      <extension base="tns:RespType"> </extension>
      <sequence>
        <extension base="tns:AccountType"> </extension>
      </sequence>
    </complexContent>
  </complexType>

  <complexType name="EventAccountSummaryType">
    <complexContent>
      <attribute name="nRecords" type="integer" use="required"/>
      <attribute name="idFirst" type="integer" use="required"/>
    </complexContent>
  </complexType>

  <!-- ===== software management related types ===== -->

  <complexType name="SoftwareUpdateType">
    <attribute name="timedUpdate" type="boolean"/>
    <attribute name="allAtOnce" type="boolean"/>
    <attribute name="hour" type="integer"/>
    <attribute name="minute" type="integer"/>
  </complexType>

  <complexType name="SoftwareUpdateRespType">
    <complexContent>
      <extension base="tns:RespType"> </extension>
      <extension base="tns:SoftwareUpdateType"> </extension>
    </complexContent>
  </complexType>

  <complexType name="SoftwareUpdateReqType">
    <complexContent>
      <extension base="tns:RespType"> </extension>
      <extension base="tns:SoftwareUpdateType"> </extension>
    </complexContent>
  </complexType>

  <complexType name="PPFirmwareUpdateType">
    <complexContent>
      <attribute name="enable" type="boolean"/>
      <attribute name="name" type="string" use="optional"/>
    </complexContent>
  </complexType>

  <complexType name="SetPPFirmwareUpdateType">
    <complexContent>
      <extension base="tns:ReqType"> </extension>
      <extension base="tns:PPFirmwareUpdateType"> </extension>
    </complexContent>
  </complexType>

  <complexType name="GetSetPPFirmwareUpdateRespType">
    <complexContent>
      <extension base="tns:RespType"> </extension>
    </complexContent>
  </complexType>

```

```

        <extension base="tns:PPFirmwareUpdateType"> </extension>
    </complexContent>
</complexType>

<simpleType name="PPFirmwareUpdateOverviewStateType">
    <restriction base="string">
        <enumeration value="startup"> </enumeration>
        <enumeration value="idle"> </enumeration>
        <enumeration value="running"> </enumeration>
        <enumeration value="error"> </enumeration>
    </restriction>
</simpleType>

<simpleType name="PPFirmwareUpdateStatusStateType">
    <restriction base="string">
        <enumeration value="ready"> </enumeration>
        <enumeration value="pending"> </enumeration>
        <enumeration value="active"> </enumeration>
        <enumeration value="barred"> </enumeration>
        <enumeration value="error"> </enumeration>
        <enumeration value="notReachable"> </enumeration>
        <enumeration value="detached"> </enumeration>
        <enumeration value="unknown"> </enumeration>
    </restriction>
</simpleType>

<simpleType name="PPFirmwareUpdateCauseStateType">
    <restriction base="string">
        <enumeration value="none"> </enumeration>
        <enumeration value="busy"> </enumeration>
        <enumeration value="battery"> </enumeration>
        <enumeration value="barred"> </enumeration>
        <enumeration value="noMaster"> </enumeration>
        <enumeration value="crc"> </enumeration>
        <enumeration value="flash"> </enumeration>
        <enumeration value="system"> </enumeration>
        <enumeration value="version"> </enumeration>
        <enumeration value="file"> </enumeration>
        <enumeration value="container"> </enumeration>
    </restriction>
</simpleType>

<complexType name="PPFirmwareUpdateOverviewType">
    <complexContent>
        <attribute name="state" type="tns:PPFirmwareUpdateOverviewStateType" />
        <attribute name="known" type="integer" />
        <attribute name="ready" type="integer" />
        <attribute name="pending" type="integer" />
        <attribute name="active" type="integer" />
        <attribute name="barred" type="integer" />
        <attribute name="errored" type="integer" />
        <attribute name="notReachable" type="integer" />
        <attribute name="detached" type="integer" />
        <attribute name="version" type="string" />
    </complexContent>
</complexType>

<complexType name="PPFirmwareUpdateStatusType">
    <complexContent>
        <attribute name="ppn" type="integer" />
        <attribute name="state" type="tns:PPFirmwareUpdateStatusStateType" />
        <attribute name="cause" type="tns:PPFirmwareUpdateCauseStateType" />
    </complexContent>
</complexType>

```

```

        <attribute name="bytes" type="integer"/>
        <attribute name="version" type="string"/>
    </complexContent>
</complexType>

<complexType name="GetPPFirmwareUpdateOverviewRespType">
    <complexContent>
        <extension base="tns:RespType"> </extension>
        <extension base="tns:PPFirmwareUpdateOverviewType"> </extension>
    </complexContent>
</complexType>

<complexType name="GetPPFirmwareUpdateStatusType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="ppn" type="integer" use="required"/>
            <attribute name="maxRecords" type="integer" use="required"/>
        </extension>
    </complexContent>
</complexType>

<complexType name="GetPPFirmwareUpdateStatusRespType">
    <complexContent>
        <extension base="tns:RespType">
            <sequence>
                <attribute name="pFwSt" type="tns:PPFirmwareUpdateStatusType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="EventPPFirmwareUpdateStatusType">
    <complexContent>
        <attribute name="pFwSt" type="tns:PPFirmwareUpdateStatusType"/>
    </complexContent>
</complexType>

<!-- ===== system related types ===== -->

<complexType name="SystemRestartType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="resetDB" type="boolean"/>
            <attribute name="resetToFactoryDefaults" type="boolean"/>
        </extension>
    </complexContent>
</complexType>

<complexType name="NetParamsType">
    <attribute name="voiceToS" type="integer" use="required"/>
    <attribute name="sigToS" type="integer" use="required"/>
    <attribute name="ttl" type="integer" use="required"/>
    <attribute name="voiceEthPrio" type="integer"/>
    <attribute name="sigEthPrio" type="integer"/>
</complexType>

<complexType name="SetNetParamsType">
    <complexContent>
        <extension base="tns:ReqType"> </extension>
        <extension base="tns:NetParamsType"> </extension>
    </complexContent>
</complexType>

```

```

<complexType name="SetDECTPowerLimitType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <attribute name="DECTPowerLimit" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="GetSetDECTPowerLimitRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <attribute name="DECTPowerLimit" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="GetSetNetParamsRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:NetParamsType"> </extension>
  </complexContent>
</complexType>

<complexType name="RemoteAccessType">
  <complexContent>
    <attribute name="enable" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="SetRemoteAccessType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:RemoteAccessType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetRemoteAccessRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:RemoteAccessType"> </extension>
  </complexContent>
</complexType>

<complexType name="RemoteSystemDumpType">
  <complexContent>
    <attribute name="enable" type="boolean"/>
    <attribute name="hour" type="integer"/>
    <attribute name="minute" type="integer"/>
    <attribute name="url" type="tns:URLType"/>
  </complexContent>
</complexType>

<complexType name="SetRemoteSystemDumpType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:RemoteSystemDumpType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetRemoteSystemDumpRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:RemoteSystemDumpType"> </extension>
  </complexContent>
</complexType>

```

```

    </complexContent>
</complexType>

<complexType name="SystemNameType">
  <complexContent>
    <attribute name="name" type="string"/>
  </complexContent>
</complexType>

<complexType name="SetSystemNameType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:SystemNameType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetSystemNameRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:SystemNameType"> </extension>
  </complexContent>
</complexType>

<complexType name="SystemCredentialsType">
  <complexContent>
    <attribute name="username" type="string"/>
    <attribute name="password" type="string"/>
  </complexContent>
</complexType>

<complexType name="SetSystemCredentialsType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:SystemCredentialsType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetSystemCredentialsRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:SystemCredentialsType"> </extension>
  </complexContent>
</complexType>

<complexType name="SystemProvUpdTrigType">
  <complexContent>
    <attribute name="enable" type="boolean"/>
    <attribute name="hour" type="integer"/>
    <attribute name="minute" type="integer"/>
  </complexContent>
</complexType>

<complexType name="SetSystemProvUpdTrigType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:SystemProvUpdTrigType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetSystemProvUpdTrigRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
  </complexContent>

```

```

        <extension base="tns:SystemProvUpdTrigType"> </extension>
    </complexContent>
</complexType>

<complexType name="ConfigURLType">
    <complexContent>
        <attribute name="url" type="tns:URLType" />
    </complexContent>
</complexType>

<complexType name="UsedConfigURLs">
    <complexContent>
        <attribute name="url" type="tns:URLType" />
    </complexContent>
</complexType>

<complexType name="SetConfigURLType">
    <complexContent>
        <extension base="tns:ReqType"> </extension>
        <extension base="tns:ConfigURLType"> </extension>
    </complexContent>
</complexType>

<complexType name="GetSetConfigURLRespType">
    <complexContent>
        <extension base="tns:RespType"> </extension>
        <extension base="tns:ConfigURLType"> </extension>
    </complexContent>
</complexType>

<complexType name="GetUsedConfigURLRespType">
    <complexContent>
        <extension base="tns:RespType"> </extension>
        <extension base="tns:UsedConfigURLsType"> </extension>
    </complexContent>
</complexType>

<complexType name="CoreDumpURLType">
    <complexContent>
        <attribute name="url" type="tns:URLType" />
    </complexContent>
</complexType>

<complexType name="GetSetCoreDumpURLRespType">
    <complexContent>
        <extension base="tns:RespType"> </extension>
        <extension base="tns:CoreDumpURLType"> </extension>
    </complexContent>
</complexType>

<complexType name="SoftwareImageURLType">
    <complexContent>
        <attribute name="url" type="tns:URLType" />
    </complexContent>
</complexType>

<complexType name="GetSetSoftwareImageURLRespType">
    <complexContent>
        <extension base="tns:RespType"> </extension>
        <extension base="tns:SoftwareImageURLType"> </extension>
    </complexContent>
</complexType>

```



```

<complexType name="PPFirmwareURLType">
  <complexContent>
    <attribute name="url" type="tns:URLType" />
  </complexContent>
</complexType>

<complexType name="GetSetPPFirmwareURLRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns: PPFirmwareURLType" />
  </complexContent>
</complexType>

<complexType name="NTPServerType">
  <complexContent>
    <attribute name="default" type="boolean" />
    <attribute name="ntpServerName1" type="string" />
    <attribute name="ntpServerName2" type="string" />
    <attribute name="ntpServerName3" type="string" />
  </complexContent>
</complexType>

<complexType name="GetSetNTPServerRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns: NTPServerType" />
  </complexContent>
</complexType>

<complexType name="UserDataServerType">
  <complexContent>
    <extension base="tns:URLType" />
    <attribute name="useCommonFileNameOnServer" type="boolean" />
  </complexContent>
</complexType>

<complexType name="SetUserDataServerType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:URLType" />
  </complexContent>
</complexType>

<complexType name="GetSetUserDataServerRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:UserDataServerType" />
  </complexContent>
</complexType>

<complexType name="UserDeviceSyncOMMType">
  <complexContent>
    <attribute name="enable" type="boolean" />
    <attribute name="omm1DomainName" type="string" />
    <attribute name="omm2DomainName" type="string" />
    <attribute name="user" type="string" />
    <attribute name="password" type="string" />
  </complexContent>
</complexType>

```

```

<complexType name="SetUserDeviceSyncOMMType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:URLType" />
  </complexContent>
</complexType>

<complexType name="GetSetUserDeviceSyncOMMRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:UserDeviceSyncOMMType" />
  </complexContent>
</complexType>

<complexType name="PreserveUserDeviceRelation">
  <complexContent>
    <attribute name="enable" type="boolean" />
  </complexContent>
</complexType>

<complexType name="SetPreserveUserDeviceRelationType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:PreserveUserDeviceRelationType" />
  </complexContent>
</complexType>

<complexType name="GetSetPreserveUserDeviceRelationRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:PreserveUserDeviceRelationType" />
  </complexContent>
</complexType>

<complexType name="AutoDBBackupType">
  <attribute name="url" type="tns:URLType" />
</complexType>

<complexType name="SetAutoDBBackupType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="AutoDBBackupType" />
  </complexContent>
</complexType>

<complexType name="GetSetAutoDBBackupRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="AutoDBBackupType" />
  </complexContent>
</complexType>

<complexType name="AutoDBBackupFilenameType">
  <attribute name="fileName" type="string" />
</complexType>

<complexType name="GetAutoDBBackupFilenameRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="AutoDBBackupFilenameType" />
  </complexContent>
</complexType>

```

```

</complexType>

<complexType name="GetDbTransferStateType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="select" type="tns:DbTransferStateSelectType"/>
    </extension>
  </complexContent>
</complexType>

<simpleType name="DbTransferStateSelectType">
  <restriction base="string">
    <enumeration value="manualBackup"/>
    <enumeration value="manualRestore"/>
    <enumeration value="automaticBackup"/>
  </restriction>
</simpleType>

<simpleType name="DbTransferStateType">
  <restriction base="string">
    <enumeration value="idle"/>
    <enumeration value="running"/>
    <enumeration value="finished"/>
    <enumeration value="identical"/>
    <enumeration value="internal_error"/>
    <enumeration value="invalid_file"/>
    <enumeration value="invalid_param"/>
    <enumeration value="unknown"/>
  </restriction>
</simpleType>

<complexType name="GetDbTransferStateRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="select" type="tns:DbTransferStateSelectType"/>
      <attribute name="state" type="tns:DbTransferStateType"/>
      <attribute name="error" type="string"/>
      <attribute name="lastupd" type="string"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventDbTransferStateType">
  <complexContent>
    <attribute name="select" type="tns:DbTransferStateSelectType"/>
    <attribute name="state" type="tns:DbTransferStateType"/>
    <attribute name="error" type="string"/>
    <attribute name="lastupd" type="string"/>
    <attribute name="url" type="string"/>
  </complexContent>
</complexType>

<complexType name="ManualDBBackupRestoreType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="url" type="tns:URLType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="ManualDBBackupRestoreRespType">

```

```

    <complexContent>
      <extension base="tns:RespType">
        <attribute name="url" type="tns:URLType"/>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="tns:ManualUserDataImportType">
    <complexContent>
      <extension base="tns:ReqType">
        <attribute name="userId" type="string" use="optional"/>
      </extension>
    </complexContent>
  </complexType>

  <simpleType name="EventUserDataImportResultType">
    <restriction base="string">
      <enumeration value="Imported"/>
      <enumeration value="Failed"/>
    </restriction>
  </simpleType>

  <complexType name="tns:EventUserDataImportType">
    <complexContent>
      <attribute name="result" type="tns:EventUserDataImportResultType"/>
      <attribute name="filename" type="string"/>
    </complexContent>
  </complexType>

  <complexType name="EULAConfirmType">
    <attribute name="confirm" type="boolean" />
  </complexType>

  <complexType name="SetEULAConfirmType">
    <complexContent>
      <extension base="tns:ReqType" />
      <extension base="tns:EULAConfirmType" />
    </complexContent>
  </complexType>

  <complexType name="GetSetEULAConfirmRespType">
    <complexContent>
      <extension base="tns:RespType" />
      <extension base="tns:EULAConfirmType" />
    </complexContent>
  </complexType>

  <simpleType name="EventLogBufferLevelType">
    <restriction base="string">
      <enumeration value="Always"/>
      <enumeration value="GenInfo"/>
      <enumeration value="FnEnter"/>
      <enumeration value="Event"/>
      <enumeration value="MinorErr"/>
      <enumeration value="MajorErr"/>
      <enumeration value="FatalErr"/>
    </restriction>
  </simpleType>

  <complexType name="EventLogBufferEntryType">
    <attribute name="id" type="integer" />
    <attribute name="time" type="string" />
  </complexType>

```

```

    <attribute name="level" type="tns:EventLogBufferLevelType" />
    <attribute name="name" type="string" />
    <attribute name="count" type="integer" />
    <attribute name="msg" type="string" />
</complexType>

<complexType name="GetEventLogBufferType">
    <complexContent>
        <extension base="tns:ReqType" /> </extension>
        <extension base="tns:GetRecordType" /> </extension>
    </complexContent>
</complexType>

<complexType name="GetEventLogBufferRespType">
    <complexContent>
        <extension base="tns:RespType">
            <attribute name="id" type="boolean" />
            <attribute name="eof" type="boolean" />
            <sequence>
                <attribute name="e" type="tns:EventLogBufferEntryType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="EventFlashMemUsageRespType">
    <complexContent>
        <extension base="tns:RespType">
            <attribute name="level" type="integer"/>
        </extension>
    </complexContent>
</complexType>

<complexType name="SNMPType">
    <complexContent>
        <attribute name="readCommunity" type="string"/>
        <attribute name="contact" type="string"/>
        <attribute name="enableTraps" type="boolean"/>
        <attribute name="trapCommunity" type="string"/>
        <attribute name="trapHostAddr" type="string"/>
    </complexContent>
</complexType>

<complexType name="SetSNMPType">
    <complexContent>
        <extension base="tns:ReqType" /> </extension>
        <extension base="tns:SNMPType" /> </extension>
    </complexContent>
</complexType>

<complexType name="GetSetSNMPRespType">
    <complexContent>
        <extension base="tns:RespType" /> </extension>
        <extension base="tns:SNMPType" /> </extension>
    </complexContent>
</complexType>

<complexType name="SpecialBrandingType">
    <complexContent>
        <attribute name="key" type="string"/>
        <attribute name="url" type="tns:URLType"/>
    </complexContent>

```

```

</complexType>

<complexType name="SetSpecialBrandingType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:SpecialBrandingType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetSpecialBrandingRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:SpecialBrandingType"> </extension>
  </complexContent>
</complexType>

<complexType name="SyslogServerType">
  <complexContent>
    <attribute name="enable" type="boolean"/>
    <attribute name="forward" type="boolean"/>
    <attribute name="ipAddr" type="string"/>
    <attribute name="port" type="integer"/>
  </complexContent>
</complexType>

<complexType name="SetSyslogServerType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:SyslogServerType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetSyslogServerRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:SyslogServerType"> </extension>
  </complexContent>
</complexType>

<complexType name="TimeZoneType">
  <complexContent>
    <attribute name="id" type="string"/>
  </complexContent>
</complexType>

<complexType name="SetTimeZoneType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:TimeZoneType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetTimeZoneRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:TimeZoneType"> </extension>
  </complexContent>
</complexType>

<complexType name="TimeZoneTypeType">
  <complexContent>
    <attribute name="id" type="string"/>
  </complexContent>

```

```

    <attribute name="name" type="string"/>
    <attribute name="stdOffset" type="integer"/>
    <attribute name="stdMonth" type="integer"/>
    <attribute name="stdDay" type="integer"/>
    <attribute name="stdDoW" type="integer"/>
    <attribute name="stdWoM" type="integer"/>
    <attribute name="stdHour" type="integer"/>
    <attribute name="stdMin" type="integer"/>
    <attribute name="dstOffset" type="integer"/>
    <attribute name="dstMonth" type="integer"/>
    <attribute name="dstDay" type="integer"/>
    <attribute name="dstDoW" type="integer"/>
    <attribute name="dstWoM" type="integer"/>
    <attribute name="dstHour" type="integer"/>
    <attribute name="dstMin" type="integer"/>
  </complexContent>
</complexType>

<complexType name="TimeZoneTypeShortType">
  <complexContent>
    <attribute name="id" type="string"/>
    <attribute name="name" type="string"/>
    <attribute name="modified" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="TimeZoneDetailsType">
  <complexContent>
    <attribute name="zone" type="tns:TimeZoneTypeType"/>
  </complexContent>
</complexType>

<complexType name="SetTimeZoneDetailsType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:TimeZoneDetailsType" />
  </complexContent>
</complexType>

<complexType name="GetSetTimeZoneDetailsRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:TimeZoneDetailsType" />
  </complexContent>
</complexType>

<complexType name="GetTimeZoneListRespType">
  <complexContent>
    <extension base="tns:RespType">
      <sequence>
        <attribute name="zone" type="tns:TimeZoneTypeShortType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventTimeZoneListType">
  <complexContent>
    <attribute name="zone" type="tns:TimeZoneTypeShortType"/>
  </complexContent>
</complexType>

```

```

<!-- ===== DECT related types ===== -->

<complexType name="PARKType">
  <complexContent>
    <attribute name="park" type="string"/>
  </complexContent>
</complexType>

<complexType name="SetPARKType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:PARKType" />
  </complexContent>
</complexType>

<complexType name="GetSetPARKRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:PARKType" />
  </complexContent>
</complexType>

<complexType name="SARIType">
  <complexContent>
    <attribute name="sari" type="string"/>
  </complexContent>
</complexType>

<complexType name="SetSARIType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:SARIType" />
  </complexContent>
</complexType>

<complexType name="GetSetSARIRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:SARIType" />
  </complexContent>
</complexType>

<simpleType name="PARKFromServerResultType">
  <restriction base="string">
    <enumeration value="Success" />
    <enumeration value="RequestAlreadyStarted" />
    <enumeration value="NoOwnMACAddress" />
    <enumeration value="FileTransferError" />
    <enumeration value="ServerResponseError" />
    <enumeration value="InvalidPARK" />
    <enumeration value="WrongMAC" />
    <enumeration value="WrongChecksum" />
    <enumeration value="MissingParam" />
    <enumeration value="FileError" />
  </restriction>
</simpleType>

<complexType name="EventPARKFromServerResultType">
  <complexContent>
    <attribute name="result" type="tns:PARKFromServerResultType" use="required"/>
  </complexContent>
</complexType>

```



```

<complexType name="DECTAuthCodeType">
  <attribute name="ac" type="string"/>
</complexType>

<complexType name="SetDECTAuthCodeType">
  <complexContent>
    <extension base="tns:ReqType" /> </extension>
    <extension base="tns:DECTAuthCodeType" use="required" /> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetDECTAuthCodeRespType">
  <complexContent>
    <extension base="tns:RespType" /> </extension>
    <extension base="tns:DECTAuthCodeType" /> </extension>
  </complexContent>
</complexType>

<complexType name="DECTEncryptionType">
  <attribute name="enable" type="boolean"/>
</complexType>

<complexType name="SetDECTEncryptionType">
  <complexContent>
    <extension base="tns:ReqType" /> </extension>
    <extension base="tns:DECTEncryptionType" use="required" /> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetDECTEncryptionRespType">
  <complexContent>
    <extension base="tns:RespType" /> </extension>
    <extension base="tns:DECTEncryptionType" use="required" /> </extension>
  </complexContent>
</complexType>

<simpleType name="DECTRegDomainType">
  <restriction base="string">
    <enumeration value="None" />
    <enumeration value="EMEA" />
    <enumeration value="US" />
    <enumeration value="Brazil" />
    <enumeration value="Taiwan" />
    <enumeration value="Radiol910_1927MHz250mW" />
  </restriction>
</simpleType>

<complexType name="SetDECTRegDomainType">
  <complexContent>
    <extension base="tns:ReqType" /> </extension>
    <extension base="tns:DECTRegDomainType" use="required" /> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetDECTRegDomainRespType">
  <complexContent>
    <extension base="tns:RespType" /> </extension>
    <extension base="tns:DECTRegDomainType" use="required" /> </extension>
  </complexContent>
</complexType>

```

```

<simpleType name="DECTSubscriptionModeType">
  <restriction base="string">
    <enumeration value="Wildcard"/>
    <enumeration value="Configured"/>
    <enumeration value="Off"/>
  </restriction>
</simpleType>

<complexType name="SetDECTSubscriptionModeType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="mode" type="tns:DECTSubscriptionModeType" use="required"/>
      <attribute name="timeout" use="optional">
        <simpleType>
          <restriction base="integer">
            <minInclusive value="1"/>
            <maxInclusive value="60"/>
          </restriction>
        </simpleType>
      </attribute>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetSetDECTSubscriptionModeRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:DECTSubscriptionModeType" use="required"> </extension>
  </complexContent>
</complexType>

<simpleType name="DECTPagingAreaSizeValueType">
  <restriction base="integer">
    <enumeration value="16"/>
    <enumeration value="32"/>
    <enumeration value="64"/>
    <enumeration value="128"/>
    <enumeration value="256"/>
  </restriction>
</simpleType>

<complexType name="DECTPagingAreaSizeType">
  <attribute name="size" type="tns:DECTPagingAreaSizeValueType"/>
</complexType>

<complexType name="SetDECTPagingAreaSizeType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:DECTPagingAreaSizeType" use="required"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetDECTPagingAreaSizeRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:DECTPagingAreaSizeType" use="required"> </extension>
  </complexContent>
</complexType>

<complexType name="DevAutoCreateType">
  <attribute name="enable" type="boolean"/>
</complexType>

```

```

<complexType name="SetDevAutoCreateType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:DevAutoCreateType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetDevAutoCreateRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:DevAutoCreateType"> </extension>
  </complexContent>
</complexType>

<complexType name="PpBootTextType">
  <attribute name="enable" type="boolean"/>
  <attribute name="text" type="string"/>
</complexType>

<complexType name="DECTPpSettingsType">
  <attribute name="bootTextHeadline" type="tns:PpBootTextType string"/>
  <attribute name="bootTextStartup" type="tns:PpBootTextType string"/>
  <attribute name="dialByNumberOnly" type="boolean"/>
</complexType>

<complexType name="SetDECTPpSettingsType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:DECTPpSettingsType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetDECTPpSettingsRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:DECTPpSettingsType"> </extension>
  </complexContent>
</complexType>

<complexType name="RestrictedSubscriptionDurationType">
  <attribute name="restrictedSubscrDur" type="boolean"/>
</complexType>

<complexType name="SetRestrictedSubscriptionDurationType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <attribute name="restrictedSubscrDur" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="GetSetRestrictedSubscriptionDurationRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <attribute name="restrictedSubscrDur" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="SysVoiceboxNumType">
  <complexContent>
    <attribute name="voiceboxNum" type="string"/>
  </complexContent>

```

```

</complexType>

<complexType name="SetSysVoiceboxNumType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:SysVoiceboxNumType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetSysVoiceboxNumRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:SysVoiceboxNumType"> </extension>
  </complexContent>
</complexType>

<simpleType name="SysToneSchemeTypeType">
  <restriction base="integer">
    <enumeration value="AU"/>
    <enumeration value="AT"/>
    <enumeration value="BE"/>
    <enumeration value="BY"/>
    <enumeration value="BR"/>
    <enumeration value="CH"/>
    <enumeration value="CZ"/>
    <enumeration value="DK"/>
    <enumeration value="ES"/>
    <enumeration value="EE"/>
    <enumeration value="FI"/>
    <enumeration value="FR"/>
    <enumeration value="GB"/>
    <enumeration value="DE"/>
    <enumeration value="HU"/>
    <enumeration value="IT"/>
    <enumeration value="LT"/>
    <enumeration value="LV"/>
    <enumeration value="NL"/>
    <enumeration value="NO"/>
    <enumeration value="PL"/>
    <enumeration value="RU"/>
    <enumeration value="SK"/>
    <enumeration value="SE"/>
    <enumeration value="TW"/>
    <enumeration value="UA"/>
    <enumeration value="US"/>
  </restriction>
</simpleType>

<complexType name="SetSysToneSchemeType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:SysToneSchemeTypeType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetSysToneSchemeRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:SysToneSchemeType"> </extension>
  </complexContent>
</complexType>

```

```

<!-- ===== SIP and PBX related types ===== -->

<simpleType name="callerDeterminationType">
  <restriction base="string">
    <enumeration value="P-Assert-Identity"/>
    <enumeration value="From/To"/>
  </restriction>
</simpleType>

<simpleType name="semiAttendedTransferModeType">
  <restriction base="string">
    <enumeration value="Blind"/>
    <enumeration value="Attended"/>
  </restriction>
</simpleType>

<simpleType name="contactMatchingModeType">
  <restriction base="string">
    <enumeration value="Uri"/>
    <enumeration value="IpOnly"/>
    <enumeration value="IpAndUsername"/>
    <enumeration value="UsernameOnly"/>
  </restriction>
</simpleType>

<complexType name="AdvancedSIPType">
  <attribute name="mwiSubscription" type="boolean"/>
  <attribute name="userAgentInfo" type="boolean"/>
  <attribute name="dialTerminator" type="boolean"/>
  <attribute name="regFailedRetryTimer" type="integer"/>
  <attribute name="regTimeoutRetryTimer" type="integer"/>
  <attribute name="transactionTimer" type="integer"/>
  <attribute name="blacklistTimeout" type="integer"/>
  <attribute name="callerDetermination" type="tns:callerDeterminationType" use="optional"/>
  <attribute name="multipleRing" type="boolean"/>
  <attribute name="semiAttendedTransferMode" type="tns:semiAttendedTransferModeType" use="optional"/>
  <attribute name="referToWithReplaces" type="boolean"/>
  <attribute name="xAastraId" type="boolean"/>
  <attribute name="userAgentCompatibility" type="boolean"/>
  <attribute name="CallRejectStateCodeDev" type="integer"/>
  <attribute name="CallRejectStateCodeUsr" type="integer"/>
  <attribute name="sessionTimer" type="integer"/>
  <attribute name="incomingCallTimeout" type="integer"/>
  <attribute name="contactMatching" type="tns:contactMatchingModeType"/>
</complexType>

<complexType name="BackupSIPType">
  <attribute name="failoverActive" type="boolean"/>
  <attribute name="failoverTime" type="integer"/>
  <attribute name="secondaryProxyServer" type="string"/>
  <attribute name="secondaryProxyPort" type="integer"/>
  <attribute name="secondaryRegServer" type="string"/>
  <attribute name="secondaryRegPort" type="integer"/>
  <attribute name="secondaryOutboundProxyServer" type="string"/>
  <attribute name="secondaryOutboundProxyPort" type="integer"/>
  <attribute name="tertiaryProxyServer" type="string"/>
  <attribute name="tertiaryProxyPort" type="integer"/>
  <attribute name="tertiaryRegServer" type="string"/>
  <attribute name="tertiaryRegPort" type="integer"/>
  <attribute name="tertiaryOutboundProxyServer" type="string"/>
  <attribute name="tertiaryOutboundProxyPort" type="integer"/>
</complexType>

```

```

<complexType name="SetAdvancedSIType">
  <complexContent>
    <extension base="tns:ReqType" /> </extension>
    <extension base="tns:AdvancedSIType" use="optional" /> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetAdvancedSIPRespType">
  <complexContent>
    <extension base="tns:RespType" /> </extension>
    <extension base="tns:AdvancedSIType" use="required" /> </extension>
  </complexContent>
</complexType>

<complexType name="SetBackupSIType">
  <complexContent>
    <extension base="tns:ReqType" /> </extension>
    <extension base="tns:BackupSIType" use="optional" /> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetBackupSIPRespType">
  <complexContent>
    <extension base="tns:RespType" /> </extension>
    <extension base="tns:BackupSIType" use="required" /> </extension>
  </complexContent>
</complexType>

<simpleType name="TransportProtType">
  <restriction base="string">
    <enumeration value="UDP" />
    <enumeration value="TCP" />
    <enumeration value="UDPAndTCP" />
    <enumeration value="TLS" />
    <enumeration value="PersistentTLS" />
  </restriction>
</simpleType>

<complexType name="BasicSIPSetType">
  <attribute name="proxyServer" type="string" />
  <attribute name="proxyPort" type="integer" />
  <attribute name="outboundProxyServer" type="string" />
  <attribute name="outboundProxyPort" type="integer" />
  <attribute name="regServer" type="string" />
  <attribute name="regPort" type="integer" />
  <attribute name="regPeriod" type="integer" />
  <attribute name="transportProt" type="tns:TransportProtType" />
  <attribute name="sipRegisterCheckNum" type="string" />
  <attribute name="gruu" type="boolean" />
</complexType>

<complexType name="BasicSIPGetType">
  <extension base="tns:BasicSIPSetType" /> </extension>
  <attribute name="sipRegisterCheckNum" use="optional" type="string" />
</complexType>

<complexType name="SetBasicSIType">
  <complexContent>
    <extension base="tns:ReqType" /> </extension>
    <extension base="tns:BasicSIPSetType" use="optional" /> </extension>
  </complexContent>

```

```

</complexType>

<complexType name="GetSetBasicSIPRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:BasicSIPGetType" use="required"> </extension>
  </complexContent>
</complexType>

<complexType name="SecureSIPSetType">
  <attribute name="proxyServer" type="string"/>
  <attribute name="proxyPort" type="integer"/>
  <attribute name="regServer" type="string"/>
  <attribute name="regPort" type="integer"/>
  <attribute name="regPeriod" type="integer"/>
</complexType>

<complexType name="SecureSIPType">
  <attribute name="keepAliveTimeoutEnable" use="optional" type="boolean"/>
  <attribute name="timeout" use="optional" type="integer"/>
  <attribute name="sendSipsOverTLS" use="optional" type="boolean"/>
  <attribute name="authenticationTLS" use="optional" type="boolean"/>
  <attribute name="commonNameValidationTLS" use="optional" type="boolean"/>
</complexType>

<complexType name="SetSecureSIPType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:SecureSIPType"/>
  </complexContent>
</complexType>

<complexType name="GetSetSecureSIPRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:SecureSIPType" use="required"> </extension>
  </complexContent>
</complexType>

<complexType name="CertificateType">
  <complexContent>
    <attribute name="key" type="string"/>
  </complexContent>
</complexType>

<complexType name="CertificateListType">
  <complexContent>
    <sequence>
      <attribute name="certificate" type="tns:CertificateType"/>
    </sequence>
  </complexContent>
</complexType>

<complexType name="SetOMMCertificateType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <attribute name="localCertificates" type="tns:CertificateListType"/>
    <attribute name="privateKey" type="tns:CertificateListType"/>
    <attribute name="privateKeyPassword" type="string" use="optional"/>
  </complexContent>
</complexType>

```

```

<complexType name="EventOMMCertificateCnfType">
  <complexContent>
    <attribute name="nLocalCertificates" type="integer"/>
    <attribute name="nPrivateKeys" type="integer"/>
    <attribute name="privateKeyPassword" type="string" use="optional"/>
  </complexContent>
</complexType>

<complexType name="GetSetOMMCertificateRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <attribute name="nLocalCertificates" type="integer"/>
    <attribute name="nPrivateKeys" type="integer"/>
    <attribute name="privateKeyPassword" type="string" use="optional"/>
  </complexContent>
</complexType>

<complexType name="GetSetOMMRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:OMMType" use="required" />
  </complexContent>
</complexType>

<complexType name="SetSecureOMMCertificateType">
  <complexContent>
    <extension base="tns:ReqType" />
    <attribute name="localCertificates" type="tns:CertificateListType"/>
    <attribute name="privateKey" type="tns:CertificateListType"/>
    <attribute name="privateKeyPassword" type="string" use="optional"/>
  </complexContent>
</complexType>

<complexType name="GetSetSecureOMMCertificateRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <attribute name="nLocalCertificates" type="integer"/>
    <attribute name="nPrivateKeys" type="integer"/>
    <attribute name="localCertificates" type="tns:CertificateListType" use="optional"/>
    <attribute name="privateKeys" type="tns:CertificateListType" use="optional"/>
    <attribute name="privateKeyPassword" type="string" use="optional"/>
  </complexContent>
</complexType>

<complexType name="SecureOMMCertificateServerImportType">
  <complexContent>
    <attribute name="enable" type="boolean"/>
    <attribute name="url" type="tns:URLType"/>
    <attribute name="localCertificates" type="string"/>
    <attribute name="privateKey" type="string"/>
  </complexContent>
</complexType>

<complexType name="SetSecureOMMCertificateServerImportType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:SecureOMMCertificateServerImportType" />
  </complexContent>
</complexType>

<complexType name="SecureOMMCertificateServerImportType">

```



```

    <complexContent>
      <attribute name="url" type="tns:URLType"/>
      <attribute name="localCertificates" type="string"/>
      <attribute name="privateKey" type="string"/>
    </complexContent>
  </complexType>

  <complexType name="SetSecureOMMCertificateServerImportType">
    <complexContent>
      <extension base="tns:RespType" />
      <extension base="tns:SecureOMMCertificateServerImportType" />
    </complexContent>
  </complexType>

  <complexType name="GetSetSecureOMMCertificateServerImportRespType">
    <complexContent>
      <extension base="tns:RespType" />
      <extension base="tns:SecureOMMCertificateServerImportType" />
    </complexContent>
  </complexType>

  <complexType name="SecurePROVCertificateServerImportType">
    <complexContent>
      <attribute name="enable" type="boolean"/>
      <attribute name="url" type="tns:URLType"/>
      <attribute name="trustedCertificates" type="string"/>
      <attribute name="localCertificates" type="string"/>
      <attribute name="privateKey" type="string"/>
    </complexContent>
  </complexType>

  <complexType name="SetSecurePROVCertificateServerImportType">
    <complexContent>
      <extension base="tns:RespType" />
      <extension base="tns:SecurePROVCertificateServerImportType" />
    </complexContent>
  </complexType>

  <complexType name="SecurePROVCertificateServerImportType">
    <complexContent>
      <attribute name="url" type="tns:URLType"/>
      <attribute name="trustedCertificates" type="string"/>
      <attribute name="localCertificates" type="string"/>
      <attribute name="privateKey" type="string"/>
    </complexContent>
  </complexType>

  <complexType name="SetSecurePROVCertificateServerImportType">
    <complexContent>
      <extension base="tns:RespType" />
      <extension base="tns:SecurePROVCertificateServerImportType" />
    </complexContent>
  </complexType>

  <complexType name="GetSetSecurePROVCertificateServerImportRespType">
    <complexContent>
      <extension base="tns:RespType" />
      <extension base="tns:SecurePROVCertificateServerImportType" />
    </complexContent>
  </complexType>

  <complexType name="SetSecureSIPCertificateType">

```

```

    <complexContent>
      <extension base="tns:ReqType"> </extension>
      <attribute name="trustedCertificates" type="tns:CertificateListType"/>
      <attribute name="localCertificates" type="tns:CertificateListType"/>
      <attribute name="privateKey" type="tns:CertificateListType"/>
      <attribute name="privateKeyPassword" type="string" use="optional"/>
    </complexContent>
  </complexType>

  <complexType name="EventSecureSIPCertificateCnfType">
    <complexContent>
      <attribute name="nTrustedCertificates" type="integer"/>
      <attribute name="nLocalCertificates" type="integer"/>
      <attribute name="nPrivateKeys" type="integer"/>
      <attribute name="trustedCertificates" type="tns:CertificateListType" use="optional"/>
      <attribute name="localCertificates" type="tns:CertificateListType" use="optional"/>
      <attribute name="privateKeys" type="tns:CertificateListType" use="optional"/>
      <attribute name="privateKeyPassword" type="string" use="optional"/>
    </complexContent>
  </complexType>

  <complexType name="GetSetSecureSIPCertificateRespType">
    <complexContent>
      <extension base="tns:RespType"> </extension>
      <attribute name="nTrustedCertificates" type="integer"/>
      <attribute name="nLocalCertificates" type="integer"/>
      <attribute name="nPrivateKeys" type="integer"/>
      <attribute name="trustedCertificates" type="tns:CertificateListType" use="optional"/>
      <attribute name="localCertificates" type="tns:CertificateListType" use="optional"/>
      <attribute name="privateKeys" type="tns:CertificateListType" use="optional"/>
      <attribute name="privateKeyPassword" type="string" use="optional"/>
    </complexContent>
  </complexType>

  <complexType name="SecureSIPCertificateServerImportType">
    <complexContent>
      <attribute name="enable" type="boolean"/>
      <attribute name="url" type="tns:URLType"/>
      <attribute name="trustedCertificates" type="string"/>
      <attribute name="localCertificates" type="string"/>
      <attribute name="privateKey" type="string"/>
    </complexContent>
  </complexType>

  <complexType name="SetSecureSIPCertificateServerImportType">
    <complexContent>
      <extension base="tns:RespType"> </extension>
      <extension base="tns:SecureSIPCertificateServerImportType"> </extension>
    </complexContent>
  </complexType>

  <complexType name="SecureSIPCertificateServerImportType">
    <complexContent>
      <attribute name="url" type="tns:URLType"/>
      <attribute name="trustedCertificates" type="string"/>
      <attribute name="localCertificates" type="string"/>
      <attribute name="privateKey" type="string"/>
    </complexContent>
  </complexType>

  <complexType name="SetSecureSIPCertificateServerImportType">
    <complexContent>

```

```

        <extension base="tns:RespType" /> </extension>
        <extension base="tns:SecureSIPCertificateServerImportType" /> </extension>
    </complexContent>
</complexType>

<complexType name="GetSetSecureSIPCertificateServerImportRespType">
    <complexContent>
        <extension base="tns:RespType" /> </extension>
        <extension base="tns:SecureSIPCertificateServerImportType" /> </extension>
    </complexContent>
</complexType>

<simpleType name="DTMFMethodType">
    <restriction base="string">
        <enumeration value="RFC2833" />
        <enumeration value="INFO" />
        <enumeration value="Both" />
    </restriction>
</simpleType>

<complexType name="DTMFType">
    <attribute name="outOfBand" type="boolean" />
    <attribute name="payloadType" type="integer" />
    <attribute name="method" type="tns:DTMFMethodType" />
</complexType>

<complexType name="SetDTMFType">
    <complexContent>
        <extension base="tns:ReqType" /> </extension>
        <extension base="tns:DTMFType" use="optional" /> </extension>
    </complexContent>
</complexType>

<complexType name="GetSetDTMFRespType">
    <complexContent>
        <extension base="tns:RespType" /> </extension>
        <extension base="tns:DTMFType" use="required" /> </extension>
    </complexContent>
</complexType>

<simpleType name="CodecType">
    <restriction base="string">
        <enumeration value="None" />
        <enumeration value="G.711-u-law" />
        <enumeration value="G.711-A-law" />
        <enumeration value="G.729-A" />
        <enumeration value="G.722" />
    </restriction>
</simpleType>

<complexType name="RTPType">
    <complexContent>
        <sequence>
            <element name="type" type="tns:CodecType" maxOccurs="5" />
        </sequence>
        <attribute name="portBase" type="integer" />
        <attribute name="packetTime" type="integer" />
        <attribute name="silenceSupp" type="boolean" />
        <attribute name="receiverPrecedence" type="boolean" />
        <attribute name="comfortNoisePktElim" type="boolean" />
        <attribute name="singleCodecReplyInSDP" type="boolean" />
    </complexContent>

```

```

</complexType>

<complexType name="SetRTPType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:RTPType" use="required"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetRTPRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:RTPType" use="required"> </extension>
  </complexContent>
</complexType>

<complexType name="RegistrationTrafficShapingType">
  <complexContent>

    <attribute name="maxRegistrations" type="integer"/>
    <attribute name="timeout" type="integer"/>
    <attribute name="spreadRegRenew" type="boolean"/>
    <attribute name="renewalTimer" type="integer"/>

  </complexContent>
</complexType>

<complexType name="SetRegistrationTrafficShapingType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:RegistrationTrafficShapingType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetRegistrationTrafficShapingRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:RegistrationTrafficShapingType"> </extension>
  </complexContent>
</complexType>

<complexType name="SuplServType">
  <complexContent>
    <attribute name="callForwDiv" type="boolean"/>
    <attribute name="locLineHndlg" type="boolean"/>
    <attribute name="uriSeparator" type="boolean"/>
    <attribute name="callTransferByHook" type="boolean"/>
    <attribute name="reRegisterAfterFailOver" type="boolean"/>
    <attribute name="ringingOnHold" type="boolean"/>
    <attribute name="transferByHook6xxd" type="boolean"/>
    <attribute name="releaseInfoTimerActiveCall" type="integer"/>
    <attribute name="releaseInfoTimerHoldCall" type="integer"/>
    <attribute name="releaseInfoTimerFailedCall" type="integer"/>
  </complexContent>
</complexType>

<complexType name="SetSuplServType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:SuplServType"> </extension>
  </complexContent>
</complexType>

```

```

<complexType name="GetSetSuplServRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:SuplServType" />
  </complexContent>
</complexType>

<complexType name="ConferenceServerSIType">
  <complexContent>
    <attribute name="conferenceServerType" type="tns:conferenceServerType" />
    <attribute name="conferenceServerURI" type="string" />
  </complexContent>
</complexType>

<complexType name="SetConferenceServerSIType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:ConferenceServerSIType" />
  </complexContent>
</complexType>

<complexType name="GetSetConferenceServerSIPRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:ConferenceServerSIType" />
  </complexContent>
</complexType>

<complexType name="SIPPortRangeType">
  <complexContent>
    <attribute name="startPort" type="integer" />
    <attribute name="endPort" type="integer" />
  </complexContent>
</complexType>

<complexType name="PortRangeSIType">
  <complexContent>
    <attribute name="userUdpTcp" type="tns:SIPPortRangeType" />
    <attribute name="userTls" type="tns:SIPPortRangeType" />
    <attribute name="confRoomUdpTcp" type="tns:SIPPortRangeType" />
    <attribute name="confRoomTls" type="tns:SIPPortRangeType" />
  </complexContent>
</complexType>

<complexType name="SetPortRangeSIType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:PortRangeSIType" />
  </complexContent>
</complexType>

<complexType name="GetSetPortRangeSIPRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:PortRangeSIType" />
  </complexContent>
</complexType>

<complexType name="SIPPortRangeType">
  <complexContent>
    <attribute name="startPort" type="integer" />
    <attribute name="endPort" type="integer" />
  </complexContent>

```

```

    </complexContent>
</complexType>

<complexType name="IntercomCallHandlingType">
  <complexContent>
    <attribute name="autoAnswer" type="boolean" />
    <attribute name="microphoneMute" type="boolean" />
    <attribute name="warningTone" type="boolean" />
    <attribute name="allowBargeIn" type="boolean" />
    <attribute name="pushToTalkPrefix" type="string" />
  </complexContent>
</complexType>

<complexType name="SetIntercomCallHandlingType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:IntercomCallHandlingType" />
  </complexContent>
</complexType>

<complexType name="GetSetIntercomCallHandlingRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:IntercomCallHandlingType" />
  </complexContent>
</complexType>

<!-- ===== PP related types ===== -->

<simpleType name="PPLoginVariantTypeType">
  <restriction base="string">
    <enumeration value="NUMBER" />
    <enumeration value="ID" />
  </restriction>
</simpleType>

<complexType name="PPLoginVariantType">
  <complexContent>
    <attribute name="login" type="tns:PPLoginVariantTypeType" />
  </complexContent>
</complexType>

<complexType name="SetPPLoginVariantType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:PPLoginVariantType" />
  </complexContent>
</complexType>

<complexType name="GetSetPPLoginVariantRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:PPLoginVariantType" />
  </complexContent>
</complexType>

<simpleType name="PPRelTypeType">
  <restriction base="string">
    <enumeration value="Unbound" />
    <enumeration value="Dynamic" />
    <enumeration value="Fixed" />
  </restriction>

```

```

</simpleType>

<simpleType name="DECTSubscriptionStateType">
  <restriction base="string">
    <enumeration value="No" />
    <enumeration value="Unconfirmed" />
    <enumeration value="Yes" />
  </restriction>
</simpleType>

<simpleType name="HWTypeType">
  <restriction base="string">
    <enumeration value="Unknown"/>
    <enumeration value="142d"/>
    <enumeration value="610d"/>
    <enumeration value="620d"/>
    <enumeration value="630d"/>
    <enumeration value="612d"/>
    <enumeration value="622d"/>
    <enumeration value="632d"/>
    <enumeration value="650c"/>
    <enumeration value="740cv"/>
    <enumeration value="GAP"/>
  </restriction>
</simpleType>

<complexType name="PPDevType">
  <complexContent>
    <attribute name="ppn" type="integer" />
    <attribute name="timeStamp" type="long integer" />
    <attribute name="relType" type="tns:PPRelTypeType"/>
    <attribute name="uid" type="integer" />
    <attribute name="ipei" type="string" />
    <attribute name="ac" type="string" />
    <attribute name="s" type="tns:DECTSubscriptionStateType" />
    <attribute name="uak" type="string" />
    <attribute name="encrypt" type="boolean" />
    <attribute name="capMessaging" type="boolean" />
    <attribute name="capMessagingForInternalUse" type="boolean" />
    <attribute name="capEnhLocating" type="boolean" />
    <attribute name="capBluetooth" type="boolean" />
    <attribute name="ethAddr" type="string" />
    <attribute name="hwType" type="tns:HWTypeType" />
    <attribute name="ppProfileCapability" type="boolean" />
    <attribute name="ppDefaultProfileLoaded" type="boolean" />
    <attribute name="subscribeToPARIOnly" type="boolean" />
  </complexContent>
</complexType>

<complexType name="GetPPDevType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="ppn" type="integer" use="required"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="CreatePPDevType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:PPDevType"> </extension>
  </complexContent>
</complexType>

```

```

<complexType name="DeletePPDevType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="ppn" type="integer" use="required"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="SetPPDevType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:PPDevType"> </extension>
  </complexContent>
</complexType>

<complexType name="CreateGetSetPPDevRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:PPDevType"> </extension>
  </complexContent>
</complexType>

<complexType name="EventPPDevType">
  <complexContent>
    <attribute name="user" type="tns:PPDevType"/>
    <attribute name="deleted" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="PPDevSummaryType">
  <complexContent>
    <attribute name="nRecords" type="integer" use="required"/>
    <attribute name="ppnFirst" type="integer" use="required"/>
    <attribute name="subscribed" type="integer" use="required"/>
  </complexContent>
</complexType>

<complexType name="GetPPDevSummaryRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:PPDevSummaryType"> </extension>
  </complexContent>
</complexType>

<simpleType name="PPTransactionType">
  <restriction base="string">
    <enumeration value="None"/>
    <enumeration value="LinkEstablish"/>
    <enumeration value="Release"/>
    <enumeration value="Establish"/>
    <enumeration value="PPNotFound"/>
    <enumeration value="PagingStarted"/>
    <enumeration value="ReleaseFromPP"/>
    <enumeration value="PPSetupRejected"/>
    <enumeration value="ComsRelease"/>
    <enumeration value="ComsEstablish"/>
    <enumeration value="SsFac"/>
    <enumeration value="SsRelease"/>
    <enumeration value="SsEstablish"/>
    <enumeration value="ConnHandover"/>
    <enumeration value="LocReg"/>
  </restriction>
</simpleType>

```



```

        <enumeration value="Detach"/>
        <enumeration value="Cc"/>
        <enumeration value="Ss"/>
        <enumeration value="Coms"/>
    </restriction>
</simpleType>

<simpleType name="PPRegServerType">
    <restriction base="string">
        <enumeration value="None"/>
        <enumeration value="Primary"/>
        <enumeration value="Secondary"/>
        <enumeration value="Tertiary"/>
    </restriction>
</simpleType>

<complexType name="EventPPTransactionType">
    <attribute name="trType" type="tns:PPTransactionType" use="required"/>
    <attribute name="ppn" type="integer" use="required"/>
    <attribute name="rfpId" type="integer"/>
</complexType>

<complexType name="LastPPActionType">
    <attribute name="ppn" type="integer"/>
    <attribute name="trType" type="tns:PPTransactionType"/>
    <attribute name="rfpId" type="integer"/>
    <attribute name="relTime" type="integer"/>
</complexType>

<complexType name="GetActivePPDevType">
    <complexContent>
        <extension base="tns:ReqType"> </extension>
        <extension base="tns:GetRecordType"> </extension>
    </complexContent>
</complexType>

<complexType name="GetActivePPDevRespType">
    <complexContent>
        <extension base="tns:RespType">
            <sequence>
                <attribute name="pp" type="tns:LastPPActionType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="PPUserSummaryType">
    <attribute name="nRecords" type="integer" use="required"/>
    <attribute name="uidFirst" type="integer" use="required"/>
    <attribute name="nSipRegistration" type="integer" use="required"/>
    <attribute name="nLocatable" type="integer" use="required"/>
    <attribute name="nMsgSend" type="integer" use="required"/>
    <attribute name="usersActiveMonitored" type="integer" use="required"/>
    <attribute name="usersPassiveMonitored" type="integer" use="required"/>
    <attribute name="usersWarned" type="integer" use="required"/>
    <attribute name="usersUnavailable" type="integer" use="required"/>
    <attribute name="usersEscalated" type="integer" use="required"/>
</complexType>

<complexType name="GetPPUserSummaryRespType">
    <complexContent>
        <extension base="tns:RespType"> </extension>
    </complexContent>
</complexType>

```

```

        <extension base="tns:PPUserSummaryType"> </extension>
    </complexContent>
</complexType>

<complexType name="CreateFixedPPType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="user" type="tns:PPUserType"/>
            <attribute name="pp" type="tns:PPDevType"/>
        </extension>
    </complexContent>
</complexType>

<complexType name="CreateFixedPPRespType">
    <complexContent>
        <extension base="tns:RespType">
            <attribute name="user" type="tns:PPUserType"/>
            <attribute name="pp" type="tns:PPDevType"/>
        </extension>
    </complexContent>
</complexType>

<complexType name="PPStateType">
    <complexContent>
        <attribute name="ppn" type="integer"/>
        <attribute name="onHook" type="boolean"/>
        <attribute name="silentCharging" type="boolean"/>
        <attribute name="batteryLevel" type="integer"/>
        <attribute name="swVersion" type="string"/>
        <attribute name="callState" type="tns:PPUserCallStateType"/>
        <attribute name="bluetooth" type="integer"/>
        <attribute name="registered" type="boolean"/>
        <attribute name="regServerType" type="tns:PPRegServerType"/>
        <attribute name="regServerAddr" type="string"/>
        <attribute name="regServerPort" type="integer"/>
    </complexContent>
</complexType>

<complexType name="GetPPStateType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="ppn" type="integer"/>
        </extension>
    </complexContent>
</complexType>

<complexType name="GetPPStateRespType">
    <complexContent>
        <extension base="tns:RespType"> </extension>
        <extension base="tns:PPStateType"> </extension>
    </complexContent>
</complexType>

<simpleType name="PPUserCallForwardStateType">
    <restriction base="string">
        <enumeration value="Off"> </enumeration>
        <enumeration value="Busy"> </enumeration>
        <enumeration value="NoAnswer"> </enumeration>
        <enumeration value="BusyNoAnswer"> </enumeration>
        <enumeration value="All"> </enumeration>
    </restriction>
</simpleType>

```

```

<simpleType name="PPUserLanguageType">
  <restriction base="string">
    <enumeration value="English"> </enumeration>
    <enumeration value="German"> </enumeration>
    <enumeration value="French"> </enumeration>
    <enumeration value="Spanish"> </enumeration>
    <enumeration value="Italian"> </enumeration>
    <enumeration value="Dutch"> </enumeration>
    <enumeration value="Swedish"> </enumeration>
    <enumeration value="Portuguese"> </enumeration>
    <enumeration value="Danish"> </enumeration>
    <enumeration value="Finnish"> </enumeration>
    <enumeration value="Norwegian"> </enumeration>
    <enumeration value="Czech"> </enumeration>
    <enumeration value="Slovakian"> </enumeration>
    <enumeration value="Hungarian"> </enumeration>
    <enumeration value="Russian"> </enumeration>
    <enumeration value="Turkish"> </enumeration>
    <enumeration value="Polish"> </enumeration>
    <enumeration value="Estonian"> </enumeration>
  </restriction>
</simpleType>

<simpleType name="PPUserMonitoringStateType">
  <restriction base="string">
    <enumeration value="Unknown"> </enumeration>
    <enumeration value="Available"> </enumeration>
    <enumeration value="Warning"> </enumeration>
    <enumeration value="Unavailable"> </enumeration>
    <enumeration value="Escalated"> </enumeration>
  </restriction>
</simpleType>

<simpleType name="BTSensitivityType">
  <restriction base="string">
    <enumeration value="low"> </enumeration>
    <enumeration value="medium"> </enumeration>
    <enumeration value="high"> </enumeration>
  </restriction>
</simpleType>

<simpleType name="PPUserMonitoringModeType">
  <restriction base="string">
    <enumeration value="Off"/>
    <enumeration value="Passive"/>
    <enumeration value="Active"/>
  </restriction>
</simpleType>

<simpleType name="IncominCallHandlingType">
  <restriction base="string">
    <enumeration value="Off"/>
    <enumeration value="On"/>
    <enumeration value="Global"/>
  </restriction>
</simpleType>

<complexType name="PPUserType">
  <complexContent>
    <attribute name="uid" type="integer" />
    <attribute name="ppn" type="integer" />
  </complexContent>
</complexType>

```

```

<attribute name="timeStamp" type="long integer" />
<attribute name="relType" type="tns:PPRelTypeType"/>
<attribute name="name" type="string" />
<attribute name="num" type="string" />
<attribute name="hierarchy1" type="string" />
<attribute name="hierarchy2" type="string" />
<attribute name="addId" type="string" />
<attribute name="pin" type="string" />
<attribute name="sipAuthId" type="string" />
<attribute name="sipPw" type="string" />
<attribute name="sosNum" type="string" />
<attribute name="voiceboxNum" type="string" />
<attribute name="manDownNum" type="string" />
<attribute name="forwardState" type="tns:PPUserCallForwardStateType" />
<attribute name="forwardTime" type="integer" />
<attribute name="forwardDest" type="string" />
<attribute name="lang" type="tns:PPUserLanguageType" />
<attribute name="holdRingBackTime" type="integer" />
<attribute name="autoAnswer" type="tns:IncominCallHandlingType" />
<attribute name="microphoneMute" type="tns:IncominCallHandlingType" />
<attribute name="warningTone" type="tns:IncominCallHandlingType" />
<attribute name="allowBargeIn" type="tns:IncominCallHandlingType" />
<attribute name="callWaitingDisabled" type="boolean" />
<attribute name="external" type="boolean" />
<attribute name="trackingActive" type="boolean" />
<attribute name="locatable" type="boolean" />
<attribute name="BTlocatable" type="boolean" />
<attribute name="BTsensitivity" type="tns:BTsensitivityType" />
<attribute name="locRight" type="boolean" />
<attribute name="msgRight" type="boolean" />
<attribute name="sendVcardRight" type="boolean" />
<attribute name="recvVcardRight" type="boolean" />
<attribute name="keepLocalPB" type="boolean" />
<attribute name="vip" type="boolean" />
<attribute name="sipRegisterCheck" type="boolean" />
<attribute name="allowVideoStream" type="boolean" />
<attribute name="conferenceServerType" type="tns:conferenceServerType" />
<attribute name="conferenceServerURI" type="string" />
<attribute name="monitoringMode" type="tns:PPUserMonitoringModeType" />
<attribute name="CUS" type="tns:PPUserMonitoringStateType" />
<attribute name="HAS" type="tns:PPUserMonitoringStateType" />
<attribute name="HSS" type="tns:PPUserMonitoringStateType" />
<attribute name="HRS" type="tns:PPUserMonitoringStateType" />
<attribute name="HCS" type="tns:PPUserMonitoringStateType" />
<attribute name="SRS" type="tns:PPUserMonitoringStateType" />
<attribute name="SCS" type="tns:PPUserMonitoringStateType" />
<attribute name="CDS" type="tns:PPUserMonitoringStateType" />
<attribute name="HBS" type="tns:PPUserMonitoringStateType" />
<attribute name="BTS" type="tns:PPUserMonitoringStateType" />
<attribute name="SWS" type="tns:PPUserMonitoringStateType" />
<attribute name="credentialPw" type="string" />
<attribute name="configurationDataLoaded" type="boolean" />
<attribute name="ppProfileId" type="integer" />
<attribute name="ppData" type="string" />
<attribute name="fixedSipPort" type="integer" />
<attribute name="calculatedSipPort" type="integer" />
</complexContent>
</complexType>

<complexType name="GetPPUserType">
  <complexContent>
    <extension base="tns:ReqType">

```

```

        <attribute name="uid" type="integer" use="required"/>
        <attribute name="maxRecords" type="integer" use="optional" default="1" />
    </extension>
</complexContent>
</complexType>

<complexType name="CreateSetPPUserType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="user" type="tns:PPUserType"/>
        </extension>
    </complexContent>
</complexType>

<complexType name="DeletePPUserType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="uid" type="integer" use="optional"/>
            <attribute name="num" type="string" use="optional"/>
        </extension>
    </complexContent>
</complexType>

<complexType name="EventPPUserType">
    <complexContent>
        <attribute name="user" type="tns:PPUserType"/>
        <attribute name="deleted" type="boolean"/>
    </complexContent>
</complexType>

<complexType name="GetPPUserRespType">
    <complexContent>
        <extension base="tns:RespType">
            <sequence>
                <attribute name="user" type="tns:PPUserType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="CreateSetPPUserRespType">
    <complexContent>
        <extension base="tns:RespType"> </extension>
        <extension base="tns:PPUserType"> </extension>
    </complexContent>
</complexType>

<complexType name="PPUserDevRelationType">
    <complexContent>
        <attribute name="uid" type="integer"/>
        <attribute name="relType" type="tns:PPRelTypeType"/>
    </complexContent>
</complexType>

<complexType name="SetPPUserDevRelationType">
    <complexContent>
        <extension base="tns:ReqType"> </extension>
        <extension base="tns:PPUserDevRelationType"> </extension>
    </complexContent>
</complexType>

<complexType name="SetPPUserDevRelationRespType">

```

```

    <complexContent>
      <extension base="tns:RespType"> </extension>
      <extension base="tns:PPUserDevRelationType"> </extension>
    </complexContent>
  </complexType>

  <complexType name="PPUserTrackingType">
    <complexContent>
      <attribute name="uid" type="integer"/>
      <attribute name="trackingActive" type="boolean"/>
    </complexContent>
  </complexType>

  <complexType name="SetPPUserTrackingType">
    <complexContent>
      <extension base="tns:ReqType"> </extension>
      <extension base="tns:PPUserTrackingType"> </extension>
    </complexContent>
  </complexType>

  <complexType name="SetPPUserTrackingRespType">
    <complexContent>
      <extension base="tns:RespType"> </extension>
      <extension base="tns:PPUserTrackingType"> </extension>
    </complexContent>
  </complexType>

  <!-- ===== RFP related types ===== -->

  <simpleType name="RFPHWTypeType">
    <restriction base="string">
      <enumeration value="Unknown"/>
      <enumeration value="RFP 31"/>
      <enumeration value="RFP 33"/>
      <enumeration value="RFP 41"/>
      <enumeration value="RFP 42"/>
      <enumeration value="RFP 42 US"/>
      <enumeration value="RFP 32"/>
      <enumeration value="RFP 32 US"/>
      <enumeration value="RFP 34"/>
      <enumeration value="RFP 34 US"/>
      <enumeration value="RFP 35"/>
      <enumeration value="RFP 36"/>
      <enumeration value="RFP 37"/>
      <enumeration value="RFP L31"/>
      <enumeration value="RFP L41"/>
      <enumeration value="RFP L42"/>
      <enumeration value="RFP L42 US"/>
      <enumeration value="RFP L32"/>
      <enumeration value="RFP L32 US"/>
      <enumeration value="RFP L34"/>
      <enumeration value="RFP L34 US"/>
      <enumeration value="RFP L35"/>
      <enumeration value="RFP L36"/>
      <enumeration value="RFP L37"/>
      <enumeration value="RFP L43"/>
      <enumeration value="RFP SL35"/>
      <enumeration value="RFP SL36"/>
      <enumeration value="RFP SL37"/>
      <enumeration value="RFP SL43"/>
      <enumeration value="PC ECM"/>
    </restriction>
  </simpleType>

```

```

</simpleType>

<simpleType name="RFPRadioTypeType">
  <restriction base="string">
    <enumeration value="LowTX"/>
    <enumeration value="NormalTX"/>
    <enumeration value="ConfigurableTX"/>
    <enumeration value="None"/>
  </restriction>
</simpleType>

<simpleType name="WLANChannelType">
  <restriction base="integer">
    <enumeration value="1"/>
    <enumeration value="2"/>
    <enumeration value="3"/>
    <enumeration value="4"/>
    <enumeration value="5"/>
    <enumeration value="6"/>
    <enumeration value="7"/>
    <enumeration value="8"/>
    <enumeration value="9"/>
    <enumeration value="10"/>
    <enumeration value="11"/>
    <enumeration value="12"/>
    <enumeration value="13"/>
    <enumeration value="14"/>
    <enumeration value="36"/>
    <enumeration value="40"/>
    <enumeration value="44"/>
    <enumeration value="48"/>
    <enumeration value="52"/>
    <enumeration value="56"/>
    <enumeration value="60"/>
    <enumeration value="64"/>
    <enumeration value="100"/>
    <enumeration value="104"/>
    <enumeration value="108"/>
    <enumeration value="112"/>
    <enumeration value="116"/>
    <enumeration value="120"/>
    <enumeration value="124"/>
    <enumeration value="128"/>
    <enumeration value="132"/>
    <enumeration value="136"/>
    <enumeration value="140"/>
    <enumeration value="147"/>
    <enumeration value="151"/>
    <enumeration value="155"/>
    <enumeration value="159"/>
    <enumeration value="163"/>
    <enumeration value="167"/>
    <enumeration value="171"/>
  </restriction>
</simpleType>

<simpleType name="WLANPowerType">
  <restriction base="integer">
    <enumeration value="6"/>
    <enumeration value="12"/>
    <enumeration value="25"/>
    <enumeration value="50"/>
  </restriction>

```

```

        <enumeration value="100"/>
    </restriction>
</simpleType>

<simpleType name="WLANAntennaType">
    <restriction base="integer">
        <enumeration value="1"/>
        <enumeration value="2"/>
    </restriction>
</simpleType>

<simpleType name="WLANChannelHtType">
    <restriction base="integer">
        <enumeration value="None"/>
        <enumeration value="HT20"/>
        <enumeration value="HT40MINUS"/>
        <enumeration value="HT40PLUS"/>
    </restriction>
</simpleType>

<simpleType name="RFPStateType">
    <restriction base="string">
        <enumeration value="Inactive"/>
        <enumeration value="NotSynced"/>
        <enumeration value="Searching"/>
        <enumeration value="Synced"/>
    </restriction>
</simpleType>

<complexType name="RFPType">
    <attribute name="id" type="integer"/>
    <attribute name="ethAddr" type="string"/>
    <attribute name="dectOn" type="boolean" default="false"/>
    <attribute name="wlanOn" type="boolean" default="false"/>
    <attribute name="licenseRfp" type="boolean" default="false"/>
    <attribute name="location" type="string"/>
    <attribute name="name" type="string"/>
    <attribute name="hierarchy1" type="string"/>
    <attribute name="hierarchy2" type="string"/>
    <attribute name="hierarchy3" type="string"/>
    <attribute name="hierarchy4" type="string"/>
    <attribute name="rpn" type="integer"/>
    <attribute name="pagingArea" type="integer"/>
    <attribute name="cluster" type="integer" default="1"/>
    <attribute name="preferredSync" type="boolean" />
    <attribute name="reflectiveEnv" type="boolean" />
    <attribute name="site" type="integer"/>
    <attribute name="x" type="integer" />
    <attribute name="y" type="integer" />
    <attribute name="hwType" type="tns:RFPHWTypeType"/>
    <attribute name="hwTypeLocked" type="boolean" />
    <attribute name="conferenceChannels" type="boolean" />
    <attribute name="wlanProfile" type="integer"/>
    <attribute name="wlanAntennaDiv" type="boolean" />
    <attribute name="wlanHighThroughput" type="boolean" />
    <attribute name="wlanAntenna" type="tns:WLANAntennaType"/>
    <attribute name="wlanChannel" type="tns:WLANChannelType"/>
    <attribute name="wlanPower" type="tns:WLANPowerType" />
    <attribute name="wlanChannelUsed" type="tns:WLANChannelType" />
    <attribute name="wlanHighThroughputTypeUsed" type="tns:WLANChannelHtType" />
    <attribute name="wlanPowerUsed" type="integer" />

```



```

<attribute name="connected" type="boolean" />
<attribute name="ipAddr" type="string" />
<attribute name="newSoftwareRequest" type="boolean" />
<attribute name="dectRunning" type="boolean" />
<attribute name="wlanRunning" type="boolean" />
<attribute name="ommRunning" type="boolean" />
<attribute name="ommStbRunning" type="boolean" />
<attribute name="hasWlan" type="boolean" />
<attribute name="hasEncryption" type="boolean" />
<attribute name="hasAdvancedFeatures" type="boolean" />
<attribute name="syncState" type="tns:RFPSyncStateType" />
<attribute name="swVersion" type="string"/>
<attribute name="brandingMismatch" type="boolean" />
<attribute name="versionMismatch" type="boolean" />
<attribute name="stbMismatch" type="boolean" />
<attribute name="wlanLinkNok" type="boolean" />
<attribute name="nSyncRels" type="integer" />

<attribute name="radioType" type="tns:RFPRadioTypeType"/>
<attribute name="outdoorType" type="boolean" />
<attribute name="hasFreqShift" type="boolean" />
</complexType>

<complexType name="RFPSummaryType">
  <attribute name="nRFPs" type="integer" use="required"/>
  <attribute name="idFirst" type="integer" use="required"/>
  <attribute name="nConnected" type="integer" use="required"/>
  <attribute name="wrongBrandedRFPs" type="integer" use="required"/>
  <attribute name="wrongStandbyRFPs" type="integer" use="required"/>
  <attribute name="wrongVersionedRFPs" type="integer" use="required"/>
  <attribute name="newAvailsSWRFPs" type="integer" use="required"/>
  <attribute name="DecryptedDECTRFPs" type="integer" use="required"/>
  <attribute name="DECTactivatedRFPs" type="integer" use="required"/>
  <attribute name="DECTactiveRFPs" type="integer" use="required"/>
  <attribute name="usedDECTclusters" type="integer" use="required"/>
  <attribute name="usbOverloads" type="integer" use="required"/>
  <attribute name="advancedFeaturesErrorRFPs" type="integer" use="required"/>
  <attribute name="WLANactivatedRFPs" type="integer" use="required"/>
  <attribute name="WLANrunningRFPs" type="integer" use="required"/>
  <attribute name="usedWLANprofiles" type="integer" use="required"/>
  <attribute name="usedPagingAreas" type="integer" use="required"/>
</complexType>

<complexType name="GetRFPSummaryRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:RFPSummaryType" />
  </complexContent>
</complexType>

<complexType name="GetRFPTType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:GetRecordType" />
    <attribute name="withState" type="boolean" default="false"/>
    <attribute name="withDetails" type="boolean" default="false"/>
  </complexContent>
</complexType>

<complexType name="CreateSetRFPTType">
  <complexContent>
    <extension base="tns:ReqType" />
  </complexContent>
</complexType>

```

```

        <attribute name="rfp" type="tns:RFPTType" />
    </complexContent>
</complexType>

<complexType name="CreateGetSetRFPPRespType">
    <complexContent>
        <extension base="tns:RespType" /> </extension>
        <sequence>
            <attribute name="rfp" type="tns:RFPTType" />
        </sequence>
    </complexContent>
</complexType>

<complexType name="DeleteGetRFPTType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="id" type="integer" use="required" />
        </extension>
    </complexContent>
</complexType>

<complexType name="EventRFPTType">
    <complexContent>
        <attribute name="rfp" type="tns:RFPTType" />
    </complexContent>
</complexType>

<complexType name="IPAddressType">
    <complexContent>
        <attribute name="ipAddr" type="string" />
    </complexContent>
</complexType>

<complexType name="RFPMTType">
    <attribute name="useDHCP" type="boolean" />
    <attribute name="firstIpAddr" type="string" />
    <attribute name="lastIpAddr" type="string" />
    <attribute name="useLocalCfg" type="boolean" />
    <attribute name="ipAddr" type="string" />
    <attribute name="netmask" type="string" />
    <attribute name="gateway" type="string" />
    <sequence>
        <attribute name="dnsServer" type="tns:IPAddressType" />
    </sequence>
    <attribute name="dnsDomain" type="string" />
</complexType>

<complexType name="GetRFPMTType">
    <complexContent>
        <extension base="tns:ReqType" /> </extension>
    </complexContent>
</complexType>

<complexType name="SetRFPTType">
    <complexContent>
        <extension base="tns:ReqType" />
        <attribute type="tns:RFPMTType" />
    </complexContent>
</complexType>

<complexType name="GetSetRFPPMRespType">
    <complexContent>

```

```

        <extension base="tns:RespType" > </extension>
        <extension base="tns:RFPMTType " > </extension>
    </complexContent>
</complexType>

<complexType name="GetRFPMTType">
    <complexContent>
        <extension base="tns:ReqType">
            </extension>
        </complexContent>
    </complexType>

<complexType name="RFPCCaptureType">
    <complexContent>
        <attribute name="enable" type="boolean"/>
    </complexContent>
</complexType>

<complexType name="SetRFPCCaptureType">
    <complexContent>
        <extension base="tns:ReqType"> </extension>
        <extension base="tns:RFPCCaptureType"> </extension>
    </complexContent>
</complexType>

<complexType name="EventRFPCCaptureType">
    <complexContent>
        <attribute name="deleted" type="boolean"/>
        <extension base="tns:RFPCCaptureType"> </extension>
    </complexContent>
</complexType>

<complexType name="GetSetRFPCCaptureRespType">
    <complexContent>
        <extension base="tns:RespType"> </extension>
        <extension base="tns:RFPCCaptureType"> </extension>
    </complexContent>
</complexType>

<complexType name="DeleteRFPCCaptureListElemType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="ethAddr" type="string"/>
        </extension>
    </complexContent>
</complexType>

<complexType name="RFPcapturedRFPType">
    <complexContent>
        <attribute name="ethAddr" type="string"/>
        <attribute name="ipAddr" type="string"/>
        <attribute name="hwType" type="tns:RFPHWTypeType"/>
        <attribute name="hasWlan" type="boolean"/>
        <attribute name="radioType" type="tns:RFPRadioTypeType"/>
        <attribute name="outdoorType" type="boolean"/>
        <attribute name="hasFreqShift" type="boolean"/>
    </complexContent>
</complexType>

<complexType name="GetRFPCCaptureListRespType">
    <complexContent>
        <extension base="tns:RespType">

```

```

        <attribute name="rfp" type="tns:RFPcapturedRFPTType" />
    </extension>
</complexContent>
</complexType>

<!-- ===== WLAN related types ===== -->

<complexType name="ACLEntryTypeType">
    <attribute name="name" type="string" />
    <attribute name="ethAddr" type="string" />
</complexType>

<complexType name="ACLEntryType">
    <complexContent>
        <attribute name="wlanProfile" type="integer" />
        <sequence>
            <element name="entry" type="tns:ACLEntryTypeType" />
        </sequence>
    </complexContent>
</complexType>

<complexType name="GetACLEntryType">
    <complexContent>
        <extension base="tns:ReqType">
            <element name="entry" type="tns:ACLEntryTypeType" />
            <attribute name="maxRecords" type="integer" use="optional" />
            <attribute name="wlanProfile" type="integer" use="required" />
        </extension>
    </complexContent>
</complexType>

<complexType name="CreateSetACLEntryType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="entry" type="tns:ACLEntryTypeType" use="required" />
        </extension>
    </complexContent>
</complexType>

<complexType name="DeleteACLEntryType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="wlanProfile" type="integer" use="required" />
            <attribute name="deleteAll" type="boolean" use="optional" />
            <attribute name="entry" type="tns:ACLEntryTypeType" use="optional" />
        </extension>
    </complexContent>
</complexType>

<complexType name="CreateGetSetACLEntryRespType">
    <complexContent>
        <extension base="tns:RespType" />
        <extension base="tns:ACLEntryType" />
    </complexContent>
</complexType>

<complexType name="EventACLCnfType">
    <complexContent>
        <extension base="tns:ACLEntryType">
            <attribute name="deleted" type="boolean" use="optional" />
            <attribute name="deleteAll" type="boolean" use="optional" />
        </extension>
    </complexContent>
</complexType>

```

```

    </complexContent>
</complexType>

<simpleType name="WLANProfileRateType">
  <restriction base="integer">
    <enumeration value="1000"> </enumeration>
    <enumeration value="2000"> </enumeration>
    <enumeration value="5500"> </enumeration>
    <enumeration value="6000"> </enumeration>
    <enumeration value="9000"> </enumeration>
    <enumeration value="11000"> </enumeration>
    <enumeration value="12000"> </enumeration>
    <enumeration value="18000"> </enumeration>
    <enumeration value="22000"> </enumeration>
    <enumeration value="24000"> </enumeration>
    <enumeration value="36000"> </enumeration>
    <enumeration value="48000"> </enumeration>
    <enumeration value="54000"> </enumeration>
  </restriction>
</simpleType>

<simpleType name="WLANProfileModeType">
  <restriction base="string">
    <enumeration value="B"> </enumeration>
    <enumeration value="G"> </enumeration>
    <enumeration value="BG"> </enumeration>
    <enumeration value="ABG"> </enumeration>
    <enumeration value="N"> </enumeration>
  </restriction>
</simpleType>

<simpleType name="WLANProfileQosType">
  <restriction base="string">
    <enumeration value="None"> </enumeration>
    <enumeration value="WME"> </enumeration>
    <enumeration value="WME-VLAN"> </enumeration>
    <enumeration value="WME-DiffServ"> </enumeration>
  </restriction>
</simpleType>

<simpleType name="WLANProfileProfileType">
  <restriction base="string">
    <enumeration value="RFP42"> </enumeration>
    <enumeration value="RFP43"> </enumeration>
  </restriction>
</simpleType>

<simpleType name="WLANPPProfileSecurityType">
  <restriction base="string">
    <enumeration value="Open"> </enumeration>
    <enumeration value="WEP"> </enumeration>
    <enumeration value="WPA"> </enumeration>
    <enumeration value="Radius"> </enumeration>
  </restriction>
</simpleType>

<simpleType name="WLANProfileWPAType">
  <restriction base="string">
    <enumeration value="Any"> </enumeration>
    <enumeration value="WPA1"> </enumeration>
    <enumeration value="WPA2"> </enumeration>
  </restriction>

```

```

</simpleType>

<simpleType name="WLANProfileKeyLengthType">
  <restriction base="integer">
    <enumeration value="64"> </enumeration>
    <enumeration value="128"> </enumeration>
    <enumeration value="256"> </enumeration>
  </restriction>
</simpleType>

<complexType name="WLANSSIDProfileType">
  <attribute name="ssid" type="string"/>
  <attribute name="enable" type="boolean"/>
  <attribute name="vlanTag" type="integer"/>
  <attribute name="security" type="tns:WLANPPProfileSecurityType"/>
  <attribute name="bssIsolation" type="boolean"/>
  <attribute name="useACL" type="boolean"/>
  <attribute name="hiddenSSID" type="boolean"/>
  <attribute name="keyIntval" type="integer"/>
  <attribute name="privacy" type="boolean"/>
  <attribute name="wepKey1" type="string"/>
  <attribute name="wepKey2" type="string"/>
  <attribute name="wepKey3" type="string"/>
  <attribute name="wepKey4" type="string"/>
  <attribute name="defaultWepKey" type="string"/>
  <attribute name="wepKeyHex" type="boolean"/>
  <attribute name="wpaType" type="tns:WLANProfileWPAType"/>
  <attribute name="psk" type="string"/>
  <attribute name="pskHex" type="boolean"/>
  <attribute name="useRadius" type="boolean"/>
  <attribute name="radiusAddr" type="string"/>
  <attribute name="radiusPort" type="integer"/>
  <attribute name="radiusSecret" type="string"/>
  <attribute name="keyLength" type="tns:WLANProfileKeyLengthType"/>
</complexType>

<complexType name="WLANProfileType">
  <attribute name="id" type="integer"/>
  <attribute name="enable" type="boolean"/>
  <attribute name="nRFPs" type="integer"/>
  <attribute name="beaconIntval" type="integer"/>
  <attribute name="dtimIntval" type="integer"/>
  <attribute name="rtsThreshold" type="integer"/>
  <attribute name="fragThreshold" type="integer"/>
  <attribute name="interferenceAvoidance" type="boolean"/>
  <attribute name="maxRate" type="tns:WLANProfileRateType"/>
  <attribute name="mode" type="tns:WLANProfileModeType"/>
  <attribute name="qos" type="tns:WLANProfileQosType"/>
  <attribute name="profileType" type="tns:WLANProfileProfileType"/>
  <attribute name="ssid1" type="WLANSSIDProfileType"/>
  <attribute name="ssid2" type="WLANSSIDProfileType"/>
  <attribute name="ssid3" type="WLANSSIDProfileType"/>
  <attribute name="ssid4" type="WLANSSIDProfileType"/>
</complexType>

<complexType name="CreateSetWLANProfileType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="profile" type="tns:WLANProfileType"/>
    </extension>
  </complexContent>
</complexType>

```

```

<complexType name="DeleteWLANProfileType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventWLANProfileType">
  <attribute name="profile" type="tns:WLANProfileType" use="required" />
  <attribute name="deleted" type="boolean" use="optional" />
</complexType>

<complexType name="CreateSetWLANProfileRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:WLANProfileType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetWLANProfileRespType">
  <complexContent>
    <extension base="tns:RespType">
      <sequence>
        <attribute name="profile" type="tns:WLANProfileType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="WLANClientType">
  <attribute name="ethAddr" type="string"/>
</complexType>

<complexType name="GetWLANClientsType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="ethAddr" type="string" use="required" />
      <attribute name="maxRecords" type="integer" use="optional" />
      <attribute name="rfpId" type="integer" use="required" />
    </extension>
  </complexContent>
</complexType>

<complexType name="GetWLANClientsRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="rfpId" type="integer" use="required" />
      <sequence>
        <attribute name="entry" type="tns:WLANClientType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventWLANClientType">
  <attribute name="rfpId" type="integer" use="required" />
  <attribute name="entry" type="tns:WLANClientType"/>
  <attribute name="deleted" type="boolean" />
  <attribute name="deleteAll" type="boolean" />
</complexType>

```

```

<complexType name="WLANRegDomainTypeType">
  <complexContent>
    <attribute name="name" type="string"/>
  </complexContent>
</complexType>

<complexType name="WLANRegDomainType">
  <complexContent>
    <attribute name="regDomain" type="string"/>
    <attribute name="channels2_4G" type="string"/>
    <attribute name="Channels5G" type="string"/>
  </complexContent>
</complexType>

<complexType name="SetWLANRegDomainType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:WLANRegDomainType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetWLANRegDomainListRespType">
  <complexContent>
    <extension base="tns:RespType">
      <sequence>
        <attribute name="regDomain" type="tns:WLANRegDomainTypeType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetSetWLANRegDomainRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:WLANRegDomainType"> </extension>
  </complexContent>
</complexType>

<!-- ===== feature PBX enrolment related types ===== -->

<complexType name="PBXEnrolmentType">
  <complexContent>
    <attribute name="url" type="tns:URLType"/>
    <attribute name="hourInterval" type="integer"/>
    <attribute name="triggerDateTime" type="tns:DateTimeType"/>
  </complexContent>
</complexType>

<complexType name="SetPBXEnrolmentType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:PBXEnrolmentType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetPBXEnrolmentRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:PBXEnrolmentType"> </extension>
  </complexContent>
</complexType>

```



```

<!-- ===== feature site related types ===== -->

<simpleType name="SiteSTRPType">
  <restriction base="string">
    <enumeration value="Only"/>
    <enumeration value="Preferred"/>
    <enumeration value="Disabled"/>
  </restriction>
</simpleType>

<complexType name="SiteType">
  <attribute name="id" type="integer"/>
  <attribute name="name" type="string"/>
  <attribute name="nRFPs" type="integer"/>
  <attribute name="wideband" type="boolean"/>
  <attribute name="srtp" type="SiteSTRPType"/>
  <attribute name="dectSecurity" type="boolean"/>
  <attribute name="videoStreaming" type="boolean"/>
</complexType>

<complexType name="CreateSetSiteType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="site" type="tns:SiteType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="DeleteSiteType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventSiteType">
  <attribute name="site" type="tns:SiteType" use="required" />
  <attribute name="deleted" type="boolean" use="optional" />
</complexType>

<complexType name="CreateSetSiteRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:SiteType" />
  </complexContent>
</complexType>

<complexType name="GetSiteRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="id" type="integer"/>
      <sequence>
        <attribute name="site" type="tns:SiteType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetSiteSummaryRespType">
  <complexContent>
    <extension base="tns:RespType" />
  </complexContent>
</complexType>

```

```

        <attribute name="nRecords" type="integer" use="required"/>
        <attribute name="idFirst" type="integer" use="required"/>
        <attribute name="rfpSiteNum" type="integer" use="required"/>
        <attribute name="g722SiteNum" type="integer" use="required"/>
        <attribute name="dectSecuritySiteNum" type="integer" use="required"/>
        <attribute name="videoStreamingSiteNum" type="integer" use="required"/>
        <attribute name="srtpSiteNum" type="integer" use="required"/>
    </complexContent>
</complexType>

<complexType name="EventSiteSummaryType">
    <complexContent>
        <attribute name="nRecords" type="integer" use="required"/>
        <attribute name="idFirst" type="integer" use="required"/>
        <attribute name="rfpSiteNum" type="integer" use="required"/>
        <attribute name="g722SiteNum" type="integer" use="required"/>
    </complexContent>
</complexType>

<!-- ===== feature digit treatment related types ===== -->

<simpleType name="DigitTreatmentDirectionType">
    <restriction base="string">
        <enumeration value="Incoming"/>
        <enumeration value="Outgoing"/>
        <enumeration value="Both"/>
        <enumeration value="Directory"/>
    </restriction>
</simpleType>

<!-- depending on OMM sent value "haveDigitTreatmentSimple" one of the DigitTreatmentType types -->
<complexType name="DigitTreatmentType">
    <attribute name="id" type="integer"/>
    <attribute name="externalPattern" type="string"/>
    <attribute name="internalPattern" type="string"/>
    <attribute name="directory" type="boolean"/>
    <attribute name="direction" type="tns:DigitTreatmentDirectionType"/>
    <attribute name="sites" type="boolean"/>
</complexType>

<complexType name="DigitTreatmentType">
    <attribute name="id" type="integer"/>
    <attribute name="prefix" type="string"/>
    <attribute name="substitute" type="string"/>
</complexType>

<complexType name="CreateSetDigitTreatmentType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="rule" type="tns:DigitTreatmentType"/>
        </extension>
    </complexContent>
</complexType>

<complexType name="DeleteDigitTreatmentType">
    <complexContent>
        <extension base="tns:ReqType">
            <attribute name="id" type="integer"/>
        </extension>
    </complexContent>
</complexType>

```

```

<complexType name="GetDigitTreatmentType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:GetRecordType" />
  </complexContent>
</complexType>

<complexType name="EventDigitTreatmentType">
  <attribute name="rule" type="tns:DigitTreatmentType" use="required" />
  <attribute name="deleted" type="boolean" use="optional" />
</complexType>

<complexType name="CreateSetDigitTreatmentRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:DigitTreatmentType" />
  </complexContent>
</complexType>

<complexType name="GetDigitTreatmentRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="id" type="integer" />
      <sequence>
        <attribute name="rule" type="tns:DigitTreatmentType" />
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- ===== feature FAC related types ===== -->

<simpleType name="FACFeatureType">
  <restriction base="string">
    <enumeration value="ActivateSubscription" />
    <enumeration value="ActivateWildcard" />
    <enumeration value="DeactivateSubscription" />
    <enumeration value="UserLogin" />
    <enumeration value="UserLogout" />

    <enumeration value="SystemCredentialPasswd" />
  </restriction>
</simpleType>

<complexType name="FACType">
  <attribute name="feature" type="tns:FACFeatureType" />
  <attribute name="enable" type="boolean" />
  <attribute name="fac" type="string" />
</complexType>

<complexType name="FACPrefixType">
  <attribute name="prefix" type="string" />
</complexType>

<complexType name="FACSequenceType">
  <complexContent>
    <sequence>
      <attribute name="fac" type="tns:FACType" />
    </sequence>
  </complexContent>
</complexType>

```

```

<complexType name="SetFACType">
  <complexContent>
    <extension base="tns:ReqType" /> </extension>
    <extension base="tns:FACType" /> </extension>
  </complexContent>
</complexType>

<complexType name="SetFACListType">
  <complexContent>
    <extension base="tns:ReqType" /> </extension>
    <extension base="tns:FACSequenceType" /> </extension>
  </complexContent>
</complexType>

<complexType name="SetFACPrefixType">
  <complexContent>
    <extension base="tns:ReqType" /> </extension>
    <extension base="tns:FACPrefixType" /> </extension>
  </complexContent>
</complexType>

<complexType name="GetFACListRespType">
  <complexContent>
    <extension base="tns:RespType" /> </extension>
    <extension base="tns:FACSequenceType" /> </extension>
  </complexContent>
</complexType>

<complexType name="SetFACListRespType">
  <complexContent>
    <extension base="tns:RespType" /> </extension>
    <extension base="tns:FACSequenceType" /> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetFACPrefixRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="fac" type="tns:FACPrefixType"/>
    </extension>
  </complexContent>
</complexType>

<!-- ===== feature corporate directory related types ===== -->

<simpleType name="LDAPTypeType">
  <restriction base="string">
    <enumeration value="GN"/>
    <enumeration value="SN"/>
    <enumeration value="CN"/>
  </restriction>
</simpleType>

<complexType name="LDAPType">
  <attribute name="id" type="integer"/>
  <attribute name="name" type="string"/>
  <attribute name="active" type="boolean"/>
  <attribute name="order" type="integer"/>
  <attribute name="server" type="string"/>
  <attribute name="port" type="integer"/>
  <attribute name="searchBase" type="string"/>
  <attribute name="username" type="string"/>

```

```

    <attribute name="password" type="string"/>
    <attribute name="searchType" type="tns:LDAPTypeType"/>
    <attribute name="displayType" type="tns:LDAPTypeType"/>
    <attribute name="timeout" type="integer"/>
</complexType>

<complexType name="CreateSetLDAPType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="ldap" type="tns:LDAPType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="DeleteLDAPType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetLDAPType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventLDAPType">
  <attribute name="ldap" type="tns:LDAPType" use="required" />
  <attribute name="deleted" type="boolean" use="optional" />
</complexType>

<complexType name="CreateSetLDAPRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:LDAPType" />
  </complexContent>
</complexType>

<complexType name="GetLDAPRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="id" type="integer"/>
      <sequence>
        <attribute name="ldap" type="tns:LDAPType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- ===== feature alerting related types ===== -->

<complexType name="AlarmTriggerType">
  <attribute name="id" type="integer"/>
  <attribute name="triggerId" type="string"/>
  <attribute name="fac" type="string"/>
  <attribute name="comment" type="string"/>
  <attribute name="num" type="string"/>
</complexType>

```

```

<complexType name="CreateAlarmTriggerType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="trigger" type="tns:AlarmTriggerType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="DeleteAlarmTriggerType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer" use="required"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetAlarmTriggerType">
  <complexContent>
    <extension base="tns:ReqType">
      <extension base="tns:ReqType"> </extension>
      <extension base="tns:GetRecordType"> </extension>
    </extension>
  </complexContent>
</complexType>

<complexType name="SetAlarmTriggerType">
  <complexContent>
    <extension base="tns:ReqType">
      <sequence>
        <attribute name="trigger" type="tns:AlarmTriggerType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="CreateGetSetAlarmTriggerRespType">
  <complexContent>
    <extension base="tns:RespType">
      <sequence>
        <attribute name="trigger" type="tns:AlarmTriggerType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventAlarmTriggerCnfType">
  <complexContent>
    <attribute name="alarmTrigger" type="tns:AlarmTriggerType" use="required" />
    <attribute name="deleted" type="boolean" use="optional" />
  </complexContent>
</complexType>

<complexType name="EventAlarmTriggerType">
  <complexContent>
    <attribute name="sendTime" type="unsigned integer" use="required" />
    <attribute name="id" type="unsigned integer" use="optional" default="0" />
    <attribute name="trigger" type="string" use="required" />
    <attribute name="uid" type="integer" use="optional" />
    <attribute name="ppn" type="integer" use="optional" />
    <attribute name="fromAddr" type="string" use="required" />
    <attribute name="toAddr" type="string" use="optional" />
  </complexContent>

```

```

        <attribute name="dialSuffix" type="string" use="optional" />
        <attribute name="content" type="string" use="optional" />
    </complexContent>
</complexType>

<complexType name="EventAlarmCallProgressType">
    <complexContent>
        <attribute name="ppn" type="integer" use="required" />
        <attribute name="trigger" type="string" use="required" />
        <attribute name="id" type="unsigned integer" use="optional" default="0" />
        <attribute name="destAddr" type="string" use="required" />
        <attribute name="state" type="string" use="required" />
    </complexContent>
</complexType>

<!-- ===== feature messaging related types ===== -->

<simpleType name="MessageConfirmType">
    <restriction base="string">
        <enumeration value="ReadYes" />
        <enumeration value="ReadNo" />
        <enumeration value="OrderOk" />
        <enumeration value="OrderNok" />
        <enumeration value="OrderDontKnow" />
        <enumeration value="CompleteDone" />
        <enumeration value="CompleteNotDone" />
        <enumeration value="CompleteFailed" />
    </restriction>
</simpleType>

<simpleType name="MessagePriorityType">
    <restriction base="string">
        <enumeration value="Info"/>
        <enumeration value="Low"/>
        <enumeration value="Normal"/>
        <enumeration value="High"/>
        <enumeration value="Emergency"/>
        <enumeration value="LocatingAlert"/>
    </restriction>
</simpleType>

<simpleType name="MessageFolderType">
    <restriction base="string">
        <enumeration value="Idle"/>
        <enumeration value="Inbox"/>
        <enumeration value="AddToInbox"/>
        <enumeration value="AddToPreDefined"/>
        <enumeration value="AddToOutbox"/>
    </restriction>
</simpleType>

<simpleType name="MessageEncodingType">
    <restriction base="string">
        <enumeration value="UTF-8"/>
    </restriction>
</simpleType>

<simpleType name="MessageContentType">
    <restriction base="string">
        <enumeration value="text/plain"/>
        <enumeration value="text/x-vcard"/>
    </restriction>

```

```

</simpleType>

<simpleType name="MessageProgressType">
  <restriction base="string">
    <enumeration value="Overridden"/>
    <enumeration value="DeleteOverridden"/>
    <enumeration value="Delivered"/>
    <enumeration value="DeleteDelivered"/>
    <enumeration value="PagingTimeout"/>
    <enumeration value="Rejected"/>
    <enumeration value="Busy"/>
    <enumeration value="WrongCharacterSet"/>
    <enumeration value="WrongURI"/>
    <enumeration value="TemporaryUnavailable"/>
    <enumeration value="Unknown"/>
  </restriction>
</simpleType>

<simpleType name="MessageMelodyType">
  <restriction base="string">
    <enumeration value="None"/>
    <enumeration value="MsgMelody1"/>
    <enumeration value="MsgMelody2"/>
    <enumeration value="MsgMelody3"/>
    <enumeration value="MsgMelody4"/>
    <enumeration value="MsgMelody5"/>
    <enumeration value="MsgMelody6"/>
    <enumeration value="MsgMelody7"/>
    <enumeration value="MsgMelody8"/>
    <enumeration value="MsgMelody9"/>
    <enumeration value="MsgMelody10"/>
  </restriction>
</simpleType>

<complexType name="MessageType">
  <sequence maxOccurs="3">
    <element name="confirmReq" type="tns:MessageConfirmType"/>
  </sequence>
  <attribute name="sendTime" type="unsigned integer"/>
  <attribute name="id" type="unsigned integer" default="0"/>
  <attribute name="Ppn" type="integer" />
  <attribute name="fromAddr" type="string"/>
  <attribute name="fromName" type="string"/>
  <attribute name="toAddr" type="string"/>
  <attribute name="toName" type="string"/>
  <attribute name="callbackAddr" type="string"/>
  <attribute name="callbackName" type="string"/>
  <attribute name="autoCallBack" type="boolean" default="false"/>
  <attribute name="priority" type="tns:MessagePriorityType" default="Normal"/>
  <attribute name="Folder" type="tns:MessageFolderType" default="Inbox"/>
  <attribute name="autoDelete" type="boolean"/>
  <attribute name="popUp" type="boolean" default="false" />
  <attribute name="encoding" type="tns:MessageEncodingType" default="UTF-8"/>
  <attribute name="contentType" type="tns:MessageContentType" default="text/plain"/>
  <attribute name="content" type="string"/>
  <attribute name="melody" type="tns:MessageMelodyType" use="optional"/>
  <attribute name="explicitToneSelection" type="string" use="optional"/>
  <attribute name="signallingVolume" type="integer" use="optional" />
  <attribute name="discCallOnRecv" type="boolean" use="optional" />
  <attribute name="increasingVolume" type="boolean" use="optional" />
  <attribute name="vibraCall" type="boolean" use="optional" />
  <attribute name="noInbandSignalling" type="boolean" use="optional" />

```



```

    <attribute name="textColourR" type="short integer" use="optional" />
    <attribute name="textColourG" type="short integer" use="optional" />
    <attribute name="textColourB" type="short integer" use="optional" />
    <attribute name="bgColourR" type="short integer" use="optional" />
    <attribute name="bgColourG" type="short integer" use="optional" />
    <attribute name="bgColourB" type="short integer" use="optional" />
</complexType>

<complexType name="CancelDeleteMessageType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="sendTime" type="unsigned integer" use="required"/>
      <attribute name="id" type="unsigned integer" default="0" use="required"/>
      <attribute name="toAddr" type="string" use="required"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventMessageQueueEmptyType">
  <attribute name="ppn" type="integer" use="required"/>
</complexType>

<complexType name="EventMessageProgressType">
  <attribute name="sendTime" type="unsigned integer" use="required"/>
  <attribute name="id" type="unsigned integer" default="0" use="required"/>
  <attribute name="toAddr" type="string" use="required"/>
  <attribute name="event" type="tns:MessageProgressType" use="required"/>
</complexType>

<complexType name="EventMessageConfirmationType">
  <attribute name="sendTime" type="unsigned integer" use="required"/>
  <attribute name="id" type="unsigned integer" default="0" use="required"/>
  <attribute name="fromAddr" type="string" use="required"/>
  <attribute name="toAddr" type="string" use="required"/>
  <attribute name="confTime" type="unsigned integer" use="required"/>
  <attribute name="confirmation" type="tns:MessageConfirmType" use="required"/>
</complexType>

<complexType name="MessageSendType">
  <attribute name="alwaysIndirect" type="boolean" use="optional"/>
  <sequence minOccurs="1" maxOccurs="1">
    <element name="msg" type="tns:MessageType" use="required"/>
  </sequence>
</complexType>

<complexType name="SendMessageType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:MessageSendType" use="required"> </extension>
  </complexContent>
</complexType>

<!-- ===== feature IMA related types ===== -->

<complexType name="SetIMAType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:ServerType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetIMARespType">

```

```

    <complexContent>
      <extension base="tns:RespType"> </extension>
      <extension base="tns:ServerType"> </extension>
    </complexContent>
  </complexType>

<!-- ===== feature OMP related types ===== -->

<complexType name="SetOMPURLType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <attribute name="url" type="tns:URLType"/>
  </complexContent>
</complexType>

<complexType name="GetSetOMPURLRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <attribute name="url" type="tns:URLType"/>
  </complexContent>
</complexType>

<!-- ===== feature locating related types ===== -->

<complexType name="LocationType">
  <complexContent>
    <attribute name="id" type="integer"/>
    <attribute name="rssiNew" type="integer"/>
    <attribute name="rssiOld" type="integer"/>
    <attribute name="nMultiFrame" type="unsigned integer"/>
  </complexContent>
</complexType>

<complexType name="VisibilityType">
  <complexContent>
    <attribute name="id" type="integer"/>
    <attribute name="rssiAvg" type="integer"/>
  </complexContent>
</complexType>

<complexType name="LocationSequenceType">
  <complexContent>
    <sequence>
      <extension base="tns:LocationType"> </extension>
    </sequence>
  </complexContent>
</complexType>

<complexType name="VisibilitySequenceType">
  <complexContent>
    <sequence>
      <extension base="tns:VisibilityType"> </extension>
    </sequence>
  </complexContent>
</complexType>

<complexType name="EventPositionHistoryType">
  <complexContent>
    <attribute name="ppn" type="integer"/>
    <attribute name="nMultiFrame" type="unsigned integer"/>
    <attribute name="loc" type="tns:LocationSequenceType"/>
  </complexContent>

```

```

</complexType>

<complexType name="EventPositionInfoType">
  <complexContent>
    <attribute name="ppn" type="integer"/>
    <attribute name="vis" type="tns:VisibilitySequenceType"/>
  </complexContent>
</complexType>

<complexType name="EventPositionRequestType">
  <complexContent>
    <attribute name="num" type="string"/>
    <attribute name="name" type="string"/>
    <attribute name="targetNum" type="string"/>
    <attribute name="targetName" type="string"/>
  </complexContent>
</complexType>

<complexType name="EventPositionTrackType">
  <complexContent>
    <attribute name="ppn" type="integer"/>
    <attribute name="id" type="integer"/>
  </complexContent>
</complexType>

<complexType name="RequestPositionInfoType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="ppn" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<!-- ===== feature UMO related types ===== -->

<complexType name="UserMonitoringType">
  <complexContent>
    <attribute name="startupDelay" type="integer"/>
    <attribute name="escalationDelay" type="integer"/>
    <attribute name="activityTimeout1" type="integer"/>
    <attribute name="activityTimeout2" type="integer"/>
    <attribute name="locatingEscalation" type="boolean"/>
    <attribute name="batteryThresholdValue" type="integer"/>
  </complexContent>
</complexType>

<complexType name="SetUserMonitoringType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:UserMonitoringType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetSetUserMonitoringRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:UserMonitoringType"> </extension>
  </complexContent>
</complexType>

<!-- ===== feature XML related types ===== -->

```

```

<simpleType name="XMLApplicationTypeType">
  <restriction base="string">
    <enumeration value="BuiltIn"/>
    <enumeration value="BuiltInReadOnly"/>
    <enumeration value="Dynamic"/>
    <enumeration value="CorpDir"/>
  </restriction>
</simpleType>

<complexType name="XMLApplicationType">
  <attribute name="enable" type="boolean"/>
  <attribute name="id" type="integer"/>
  <attribute name="type" type="tns:XMLApplicationTypeType"/>
  <attribute name="corpDirOrder" type="integer"/>
  <attribute name="name" type="string"/>
  <attribute name="url" type="tns:URLType"/>
</complexType>

<complexType name="CreateSetXMLApplicationType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="XMLApplication" type="tns:XMLApplicationType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="DeleteXMLApplicationType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventXMLApplicationType">
  <attribute name="XMLApplication" type="tns:XMLApplicationType" use="required" />
  <attribute name="deleted" type="boolean" use="optional" />
</complexType>

<complexType name="CreateSetXMLApplicationRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <extension base="tns:XMLApplicationType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetXMLApplicationRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="id" type="integer"/>
      <sequence>
        <attribute name="XMLApplication" type="tns:XMLApplicationType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<!-- ===== feature conference related types ===== -->

<simpleType name="ConferenceChangeType">
  <restriction base="string">
    <enumeration value="added"/>
  </restriction>
</simpleType>

```

```

        <enumeration value="deleted"/>
    </restriction>
</simpleType>

<simpleType name="MediaCodecType">
    <restriction base="string">
        <enumeration value="CodecG711_A"/>
        <enumeration value="CodecG723_53"/>
        <enumeration value="CodecG723_63"/>
        <enumeration value="CodecG729"/>
        <enumeration value="CodecG711_U"/>
        <enumeration value="CodecG722"/>
        <enumeration value="CodecH261"/>
        <enumeration value="CodecH263"/>
        <enumeration value="CodecG726_16"/>
        <enumeration value="CodecG726_24"/>
        <enumeration value="CodecG726_32"/>
        <enumeration value="CodecG726_40"/>
        <enumeration value="CodecAAL2_G726_16"/>
        <enumeration value="CodecAAL2_G726_24"/>
        <enumeration value="CodecAAL2_G726_32"/>
        <enumeration value="CodecAAL2_G726_40"/>
        <enumeration value="CodecL16"/>
        <enumeration value="CodecH264"/>
        <enumeration value="CodecInvalid"/>
    </restriction>
</simpleType>

<simpleType name="SRTPProfileType">
    <restriction base="string">
        <enumeration value="None"/>
        <enumeration value="AES128_CM_SHA1_32"/>
        <enumeration value="AES128_CM_SHA1_80"/>
        <enumeration value="NULL_SHA1_32"/>
        <enumeration value="NULL_SHA1_80"/>
    </restriction>
</simpleType>

<complexType name="ConferenceCodecType">
    <attribute name="mediaCodec" type="tns:MediaCodecType"/>
    <attribute name="mediaPt" type="integer"/>
    <attribute name="rate" type="integer"/>
</complexType>

<complexType name="ConferenceDataType">
    <attribute name="mixerId" type="integer"/>
    <attribute name="num" type="string"/>
    <attribute name="localRTP" type="integer"/>
    <attribute name="localRTCP" type="integer"/>
    <attribute name="remoteRTP" type="integer"/>
    <attribute name="remoteRTCP" type="integer"/>
    <attribute name="remoteIpAddr" type="string"/>
    <attribute name="vif" type="integer"/>
    <attribute name="vad" type="boolean"/>
    <attribute name="txCodecIdx" type="integer"/>
    <attribute name="rxCodecIdx" type="integer"/>
    <sequence>
        <attribute name="codec" type="tns:ConferenceCodecType"/>
    </sequence>
    <attribute name="srtpProfile" type="tns:SRTPProfileType"/>
    <attribute name="rxSecret" type="string"/>
    <attribute name="txSecret" type="string"/>

```

```

</complexType>

<complexType name="CreateConferenceType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="conferenceId" type="string"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="DeleteConferenceType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="mixerId" type="integer"/>
      <attribute name="localRTP" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="AddUserToConferenceType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <attribute name="conference" type="ConferenceDataType"/>
  </complexContent>
</complexType>

<complexType name="ChangeUserInConferenceType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <attribute name="conference" type="ConferenceDataType"/>
  </complexContent>
</complexType>

<complexType name="DeleteUserFromConferenceType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="mixerId" type="integer"/>
      <attribute name="localRTP" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventAddChangeUserInConferenceType">
  <complexContent>
    <attribute name="conference" type="ConferenceDataType"/>
  </complexContent>
</complexType>

<complexType name="EventRTPConferenceStreamChg">
  <complexContent>
    <attribute name="mixerId" type="integer"/>
    <attribute name="num" type="string"/>
    <attribute name="localRTP" type="integer"/>
    <attribute name="rx" type="boolean"/>
    <attribute name="tx" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="SetRTPConferenceStreamChg">
  <complexContent>
    <attribute name="mixerId" type="integer"/>
    <attribute name="localRTP" type="integer"/>
  </complexContent>
</complexType>

```

```

        <attribute name="rx" type="boolean"/>
        <attribute name="tx" type="boolean"/>
    </complexContent>
</complexType>

<complexType name="EventDeleteUserFromConferenceType">
    <complexContent>
        <attribute name="mixerId" type="integer"/>
        <attribute name="num" type="string"/>
        <attribute name="localRTP" type="integer"/>
    </complexContent>
</complexType>

<complexType name="EventCreateConferenceType">
    <complexContent>
        <attribute name="mixerId" type="integer"/>
        <attribute name="conferenceId" type="string"/>
        <attribute name="rfpId" type="integer"/>
        <attribute name="IpAddr" type="string"/>
        <attribute name="port" type="integer"/>
    </complexContent>
</complexType>

<complexType name="EventDeleteConferenceType">
    <complexContent>
        <attribute name="mixerId" type="integer"/>
        <attribute name="conferenceId" type="string"/>
    </complexContent>
</complexType>

<complexType name="EventConferenceReleaseType">
    <complexContent>
        <attribute name="sipRef" type="string"/>
        <attribute name="conferenceId" type="string"/>
    </complexContent>
</complexType>

<complexType name="EventConferenceRequestType">
    <complexContent>
        <attribute name="sipRef" type="string"/>
        <attribute name="confUsers" type="integer"/>
    </complexContent>
</complexType>

<complexType name="RequestConferenceConfirmationType">
    <complexContent>
        <attribute name="sipRef" type="string"/>
        <attribute name="conferenceId" type="string"/>
    </complexContent>
</complexType>

<complexType name="GetG729ChannelsForConferenceType">
    <complexContent>
        <extension base="tns:ReqType" />
        <attribute name="channels" type="integer"/>
    </complexContent>
</complexType>

<complexType name="GetG729ChannelsForConferenceRespType">
    <complexContent>
        <extension base="tns:RespType" />
        <attribute name="channels" type="integer"/>
    </complexContent>
</complexType>

```

```

    </complexContent>
</complexType>

<complexType name="GetReadyForConferencingRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <attribute name="ready" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="CreateConferenceRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="mixerId" type="integer"/>
      <attribute name="ipAddr" type="string"/>
      <attribute name="port" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="ConferenceRoomType">
  <attribute name="id" type="integer"/>
  <attribute name="name" type="string"/>
  <attribute name="conferenceId" type="string"/>
  <attribute name="sipAuthId" type="boolean"/>
  <attribute name="sipPw" type="string"/>
  <attribute name="fixedSipPort" type="integer" />
  <attribute name="calculatedSipPort" type="integer" />
</complexType>

<complexType name="CreateSetConferenceRoomType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="room" type="tns:ConferenceRoomType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="DeleteConferenceRoomType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventConferenceRoomCnfType">
  <attribute name="room" type="tns:ConferenceRoomType" use="optional" />
  <attribute name="id" type="integer" use="optional"/>
  <attribute name="deleted" type="boolean" use="optional" />
</complexType>

<complexType name="CreateSetConferenceRoomRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <attribute name="room" type="tns:ConferenceRoomType"/>
  </complexContent>
</complexType>

<complexType name="GetConferenceRoomRespType">
  <complexContent>
    <extension base="tns:RespType">

```



```

        <sequence>
            <attribute name="room" type="tns:ConferenceRoomType"/>
        </sequence>
    </extension>
</complexContent>
</complexType>

<complexType name="FreeConferenceChannelsType">
    <attribute name="freeChannels" type="integer"/>
    <attribute name="tChannels" type="integer"/>
</complexType>

<complexType name="GetFreeConferenceChannelsRespType">
    <complexContent>
        <extension base="tns:RespType"> </extension>
        <extension base="tns:FreeConferenceChannelsType"> </extension>
    </complexContent>
</complexType>

<!-- ===== feature Blue Tooth Beacon related types ===== -->

<complexType name="BTStatisticDataType">
    <attribute name="id" type="integer"/>
    <attribute name="rssi" type="integer"/>
</complexType>

<complexType name="BTClientStatisticDataType">
    <complexContent>
        <attribute name="ethAddr" type="string"/>
        <attribute name="ppn" type="integer" use="optional" />
        <sequence>
            <attribute name="btLocStat" type="tns:BTStatisticDataType"/>
        </sequence>
    </complexContent>
</complexType>

<complexType name="BTRfpSensType">
    <attribute name="checkpoint" type="integer"/>
    <attribute name="low" type="integer"/>
    <attribute name="medium" type="integer"/>
    <attribute name="high" type="integer"/>
</complexType>

<complexType name="BTppSensType">
    <attribute name="low" type="integer"/>
    <attribute name="medium" type="integer"/>
    <attribute name="high" type="integer"/>
</complexType>

<simpleType name="BTBStateType">
    <restriction base="string">
        <enumeration value="unplugged"/>
        <enumeration value="plugged"/>
        <enumeration value="failed"/>
    </restriction>
</simpleType>

<complexType name="BTBType">
    <attribute name="id" type="integer"/>
    <attribute name="name" type="string" />
    <attribute name="ethAddr" type="string"/>
    <attribute name="rfpId" type="integer"/>

```

```

    <attribute name="rssi" type="integer"/>
    <attribute name="hierarchy1" type="string"/>
    <attribute name="hierarchy2" type="string"/>
    <attribute name="hierarchy3" type="string"/>
    <attribute name="hierarchy4" type="string"/>
    <attribute name="calibration" type="integer"/>
    <attribute name="state" type="tns:BTBStateType"/>
    <attribute name="checkpoint" type="boolean"/>
</complexType>

<complexType name="GetBluetoothBeaconType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:GetRecordType"> </extension>
  </complexContent>
</complexType>

<complexType name="CreateSetBluetoothBeaconType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="btb" type="tns:BTBType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="CreateSetBluetoothBeaconRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="btb" type="tns:BTBType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetBluetoothBeaconRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <sequence>
      <attribute name="btb" type="tns:BTBType"/>
    </sequence>
  </complexContent>
</complexType>

<complexType name="DeleteBluetoothBeaconType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer" use="required"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventBluetoothBeaconCnfType">
  <complexContent>
    <attribute name="id" type="integer" use="required"/>
    <attribute name="btb" type="tns:BTBType"/>
  </complexContent>
</complexType>

<complexType name="EventBluetoothClientStatisticType">
  <complexContent>
    <attribute name="btClientStat" type="tns:BTClientStatisticDataType"/>
  </complexContent>
</complexType>

```

```

<complexType name="GetBluetoothClientStatisticRespType">
  <complexContent>
    <extension base="tns:RespType">
      <sequence>
        <attribute name="btClientStat" type="tns:BTClientStatisticDataType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="SetBluetoothSensitivityType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="rfpSens" type="tns:BTRfpSensType"/>
      <attribute name="ppSens" type="tns:BTppSensType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetSetBluetoothSensitivityRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="rfpSens" type="tns:BTRfpSensType"/>
      <attribute name="ppSens" type="tns:BTppSensType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventBluetoothSensitivityCnfType">
  <complexContent>
    <attribute name="rfpSens" type="tns:BTRfpSensType"/>
    <attribute name="ppSens" type="tns:BTppSensType"/>
  </complexContent>
</complexType>

<complexType name="SetBluetoothGlobalSettingsType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="active" type="boolean"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetSetBluetoothGlobalSettingsRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="active" type="boolean"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventBluetoothGlobalSettingsCnfType">
  <complexContent>
    <attribute name="active" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="BluetoothBeaconSummaryType">
  <complexContent>
    <attribute name="nRecords" type="integer"/>
    <attribute name="idFirst" type="integer"/>
  </complexContent>
</complexType>

```

```

        <attribute name="checkpointNum" type="integer"/>
        <attribute name="stateUnpluggedNum" type="integer"/>
        <attribute name="statePluggedNum" type="integer"/>
        <attribute name="stateFailedNum" type="integer"/>
    </complexContent>
</complexType>

<complexType name="GetBluetoothBeaconSummaryRespType">
    <complexContent>
        <extension base="tns:RespType" />
        <attribute base="tns:BluetoothBeaconSummaryType"/>
    </complexContent>
</complexType>

<!-- ===== feature Video Camera related types ===== -->

<simpleType name="VideoDevStateType">
    <restriction base="string">
        <enumeration value="unplugged"/>
        <enumeration value="inactive"/>
        <enumeration value="active"/>
        <enumeration value="failed"/>
    </restriction>
</simpleType>

<simpleType name="tns:VideoDevResolutionType">
    <restriction base="string">
        <enumeration value="QCIF"/>
        <enumeration value="QVGA"/>
        <enumeration value="CIF"/>
        <enumeration value="VGA"/>
        <enumeration value="SVGA"/>
    </restriction>
</simpleType>

<complexType name="PpType">
    <attribute name="ppn" type="integer"/>
</complexType>

<complexType name="VideoDevType">
    <attribute name="id" type="integer"/>
    <attribute name="name" type="string" />
    <attribute name="usbPath" type="string" />
    <attribute name="tag" type="string" />
    <attribute name="rfpId" type="integer"/>
    <attribute name="active" type="boolean"/>
    <attribute name="state" type="tns:VideoDevStateType"/>
    <attribute name="resolution" type="tns:VideoDevResolutionType"/>
    <attribute name="frameRate" type="integer"/>
    <attribute name="hierarchy1" type="string"/>
    <attribute name="hierarchy2" type="string"/>
    <attribute name="hierarchy3" type="string"/>
    <attribute name="hierarchy4" type="string"/>
    <attribute name="ppnNum" type="integer"/>
    <sequence>
        <attribute name="ppList" type="tns:PpType"/>
    </sequence>
    <attribute name="siteId" type="integer"/>
    <attribute name="siteName" type="string"/>
</complexType>

<complexType name="VideoDevLinkType">

```

```

    <attribute name="id" type="integer"/>
    <attribute name="videoStream" type="boolean"/>
    <attribute name="videoDevLink" type="string"/>
</complexType>

<complexType name="CreateSetVideoDevType">
  <complexContent>
    <extension base="tns:ReqType"/>
    <attribute name="videoDev" type="tns:VideoDevType"/>
  </complexContent>
</complexType>

<complexType name="CreateGetSetVideoDevRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <sequence>
      <attribute name="videoDev" type="tns:VideoDevType"/>
    </sequence>
  </complexContent>
</complexType>

<complexType name="DeleteGetVideoDevType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer" use="required"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventVideoDevType">
  <complexContent>
    <attribute name="videoDev" type="tns:VideoDevType"/>
  </complexContent>
</complexType>

<complexType name="GetVideoDevLinkType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <attribute name="id" type="integer"/>
    <attribute name="videoStream" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="VideoDevSummaryType">
  <complexContent>
    <attribute name="nRecords" type="integer"/>
    <attribute name="idFirst" type="integer"/>
    <attribute name="enabledNum" type="integer"/>
    <attribute name="stateActiveNum" type="integer"/>
    <attribute name="stateInactiveNum" type="integer"/>
    <attribute name="stateUnpluggedNum" type="integer"/>
    <attribute name="stateFailedNum" type="integer"/>
  </complexContent>
</complexType>

<complexType name="GetVideoDevSummaryRespType">
  <complexContent>
    <extension base="tns:RespType"> </extension>
    <attribute base="tns:VideoDevSummaryType"/>
  </complexContent>
</complexType>

```

```

<!-- ===== feature Additional Settings related types ===== -->

<complexType name="AdditionalSettingsType">
  <complexContent>
    <attribute name="truncatePpUserName" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="SetAdditionalSettingsType">
  <complexContent>
    <extension base="tns:ReqType" />
    <extension base="tns:AdditionalSettingsType" />
  </complexContent>
</complexType>

<complexType name="GetSetAdditionalSettingsRespType">
  <complexContent>
    <extension base="tns:RespType" />
    <extension base="tns:AdditionalSettingsType" />
  </complexContent>
</complexType>

<!-- ===== feature SYNC related types ===== -->

<complexType name="RFPSyncType">
  <complexContent>
    <attribute name="enable" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="RFPSyncOffsetType">
  <complexContent>
    <attribute name="id" type="integer"/>
    <attribute name="offset" type="integer"/>
    <attribute name="rssi" type="integer"/>
    <attribute name="lost" type="boolean"/>
  </complexContent>
</complexType>

<complexType name="GetRFPSyncType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventRFPSyncRelType">
  <attribute name="id" type="integer"/>
  <sequence>
    <attribute name="forward" type="tns:RFPSyncOffsetType"/>
  </sequence>
</complexType>

<complexType name="GetSetRFPSyncRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="id" type="integer"/>
      <attribute name="syncState" type="tns:RFPSyncStateType"/>
      <sequence>
        <attribute name="forward" type="tns:RFPSyncOffsetType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

```

```

        <sequence>
            <attribute name="backward" type="tns:RFPSType"/>
        </sequence>
    </extension>
</complexContent>
</complexType>

<!-- ===== RFP quality related types ===== -->

<complexType name="RFPipQualityType">
    <complexContent>
        <attribute name="id" type="integer"/>
        <attribute name="connectedTime" type="integer"/>
        <attribute name="currentRTT" type="integer"/>
        <attribute name="maxRTT" type="integer"/>
        <attribute name="count" type="integer"/>
        <attribute name="interval1" type="integer"/>
        <attribute name="interval2" type="integer"/>
        <attribute name="interval3" type="integer"/>
        <attribute name="interval4" type="integer"/>
        <attribute name="interval5" type="integer"/>
    </complexContent>
</complexType>

<complexType name="GetRFPipQualityType">
    <complexContent>
        <extension base="tns:ReqType" />
        <extension base="tns:GetRecordType" />
    </complexContent>
</complexType>

<complexType name="GetRFPipQualityRespType">
    <complexContent>
        <extension base="tns:RespType">
            <sequence>
                <attribute name="ipQuality" type="tns:RFPipQualityType"/>
            </sequence>
        </extension>
    </complexContent>
</complexType>

<complexType name="EventRFPipQualityType">
    <complexContent>
        <sequence>
            <attribute name="ipQuality" type="tns:RFPipQualityType"/>
        </sequence>
    </complexContent>
</complexType>

<complexType name="RFPMediaStreamQualityType">
    <complexContent>
        <attribute name="connects" type="integer"/>
        <attribute name="duration" type="integer"/>
        <attribute name="packetsRx" type="integer"/>
        <attribute name="octetsRx" type="integer"/>
        <attribute name="packetsTx" type="integer"/>
        <attribute name="octetsTx" type="integer"/>
        <attribute name="packetsLost" type="integer"/>
        <attribute name="maxJitter" type="integer"/>
    </complexContent>
</complexType>

```

```

<complexType name="GetRFPMediaStreamQualityType">
  <complexContent>
    <attribute name="id" type="integer"/>
  </complexContent>
</complexType>

<complexType name="GetRFPMediaStreamQualityRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="msQuality" type="tns:RFPMediaStreamQualityType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="ResetRFPMediaStreamQualityType">
  <complexContent>
    <attribute name="id" type="integer"/>
  </complexContent>
</complexType>

<complexType name="EventRFPMediaStreamQualityType">
  <complexContent>
    <attribute name="msQuality" type="tns:RFPMediaStreamQualityType"/>
  </complexContent>
</complexType>

<complexType name="RFPsSyncQualityType">
  <complexContent>
    <attribute name="id" type="integer"/>
    <attribute name="strongRels" type="integer"/>
    <attribute name="lowRels" type="integer"/>
    <attribute name="maxRSSI" type="integer"/>
    <attribute name="minRSSI" type="integer"/>
  </complexContent>
</complexType>

<complexType name="GetRFPsSyncQualityType">
  <complexContent>
    <extension base="tns:ReqType"> </extension>
    <extension base="tns:GetRecordType"> </extension>
  </complexContent>
</complexType>

<complexType name="GetRFPsSyncQualityRespType">
  <complexContent>
    <extension base="tns:RespType">
      <sequence>
        <attribute name="syncQuality" type="tns:RFPsSyncQualityType"/>
      </sequence>
    </extension>
  </complexContent>
</complexType>

<complexType name="EventRFPsSyncQualityType">
  <complexContent>
    <sequence>
      <attribute name="syncQuality" type="tns:RFPsSyncQualityType"/>
    </sequence>
  </complexContent>
</complexType>

<!-- ===== statistic related types ===== -->

```



```

<complexType name="GetRFPStatisticType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer"/>
      <attribute name="maxRecords" type="integer"/>
      <attribute name="recordSet" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="RFPStatDataType">
  <attribute name="id" type="integer"/>
  <attribute name="counter" type="string"/>
</complexType>

<complexType name="RFPStatHeadType">
  <attribute name="numElemPerRec" type="integer"/>
  <attribute name="recordSets" type="integer"/>
  <attribute name="resolution" type="string"/>
</complexType>

<complexType name="RFPStatNameType">
  <attribute name="elemId" type="integer"/>
  <attribute name="group" type="string"/>
  <attribute name="name" type="string"/>
</complexType>

<complexType name="GetRFPStatisticRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="rfpStatData" type="tns:RFPStatDataType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetRFPStatisticConfigRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="rfpStatHead" type="tns:RFPStatHeadType"/>
      <attribute name="rfpStatName" type="tns:RFPStatNameType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="ResetRFPStatisticType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="id" type="integer"/>
      <attribute name="elemId" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="SysStatNameType">
  <attribute name="elemId" type="integer"/>
  <attribute name="type" type="string"/>
  <attribute name="name" type="string"/>
  <attribute name="resolution" type="string"/>
  <attribute name="duration" type="integer"/>
  <attribute name="numOfRecords" type="integer"/>
  <attribute name="sum" type="integer"/>

```

```

    <attribute name="unit" type="string"/>
    <attribute name="numOfFreqDistChannels" type="integer"/>
</complexType>

<complexType name="GetSysStatisticConfigConfigRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="sysStatName" type="tns:SysStatNameType"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetSysStatisticOccurrenceType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="elemId" type="integer"/>
      <attribute name="firstRecord" type="integer"/>
      <attribute name="maxRecords" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetSysStatisticOccurrenceRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="elemId" type="integer"/>
      <attribute name="age" type="integer"/>
      <attribute name="sum" type="integer"/>
      <attribute name="firstRecord" type="integer"/>
      <attribute name="counter" type="integer list"/>
      <attribute name="currMeasuredTime" type="integer"/>
      <attribute name="oldMeasuredTime" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetSysStatisticMinMaxType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="elemId" type="integer"/>
      <attribute name="type" type="string"/>
      <attribute name="firstRecord" type="integer"/>
      <attribute name="maxRecords" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetSysStatisticMinMaxRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="elemId" type="integer"/>
      <attribute name="type" type="string"/>
      <attribute name="firstRecord" type="integer"/>
      <attribute name="counter" type="string"/>
      <attribute name="occurrence" type="integer"/>
      <attribute name="currMeasuredTime" type="integer"/>
      <attribute name="oldMeasuredTime" type="integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetSysStatisticMinMaxRecordType">

```

```

    <complexContent>
      <extension base="tns:ReqType">
        <attribute name="elemId" type="integer"/>
        <attribute name="firstRecord" type="integer"/>
        <attribute name="maxRecords" type="integer"/>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="SysRecHeaderType">
    <attribute name="elemId" type="integer"/>
    <attribute name="counterName" type="string"/>
    <attribute name="firstRecord" type="integer"/>
    <attribute name="currMeasuredTime" type="integer"/>
    <attribute name="oldMeasuredTime" type="integer"/>
  </complexType>

  <complexType name="SysRecHeaderType">
    <attribute name="rec" type="integer"/>
    <attribute name="cnt" type="string"/>
  </complexType>

  <complexType name="GetSysStatisticMinMaxRecordRespType">
    <complexContent>
      <extension base="tns:RespType">
        <attribute name="sysRecHeader" type="tns:SysRecHeaderType"/>
        <attribute name="sysRec" type="tns:SysRecHeaderType"/>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="GetSysStatisticMinMaxSummaryType">
    <complexContent>
      <extension base="tns:ReqType">
        <attribute name="elemId" type="integer"/>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="GetSysStatisticMinMaxSummaryRespType">
    <complexContent>
      <extension base="tns:RespType">
        <attribute name="elemId" type="integer"/>
        <attribute name="min" type="integer"/>
        <attribute name="max" type="integer"/>
        <attribute name="sum" type="integer"/>
        <attribute name="occurrence" type="integer"/>
        <attribute name="freqDistUnits" type="integer list"/>
        <attribute name="freqDistValues" type="integer list"/>
      </extension>
    </complexContent>
  </complexType>

  <complexType name="ResetSysStatisticType">
    <complexContent>
      <extension base="tns:ReqType">
        <attribute name="elemId" type="integer"/>
      </extension>
    </complexContent>
  </complexType>

  <!-- ===== PP default profile related types ===== -->

```

```

<complexType name="PpProfileType">
  <complexContent>
    <attribute name="id" type="integer" />
    <attribute name="name" type="string" />
    <attribute name="ppData" type="string"/>
    <attribute name="timeStamp" type="long integer" />
  </complexContent>
</complexType>

<complexType name="SetPpProfileType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="ppProfile" type="tns:PpProfileType" />
    </extension>
  </complexContent>
</complexType>

<complexType name="GetPpProfileType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="Id" type="integer" use="optional" />
    </extension>
  </complexContent>
</complexType>

<complexType name="EventPpProfileType">
  <complexContent>
    <attribute name="delete" type="boolean" use="optional" />
    <attribute name="ppProfile" type="tns:PpProfileType" use="optional" />
  </complexContent>
</complexType>

<complexType name="GetSetPpProfileRespType">
  <complexContent>
    <attribute name="ppProfile" type="tns:PpProfileType" use="optional" />
  </complexContent>
</complexType>

<!-- ===== file transfer related types ===== -->

<complexType name="GetFileType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="name" type="string"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="GetFileRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="name" type="string"/>
      <attribute name="offset" type="integer"/>
      <attribute name="data" type="string"/>
      <attribute name="size" type="integer"/>
      <attribute name="eof" type="boolean"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="PutFileType">

```

```

    <complexContent>
      <extension base="tns:ReqType">
        <attribute name="name" type="string"/>
        <attribute name="offset" type="integer"/>
        <attribute name="data" type="string"/>
        <attribute name="eof" type="boolean"/>
      </extension>
    </complexContent>
  </complexType>
</complexType>

<!-- ===== Ping related types ===== -->

<complexType name="PingRespType">
  <complexContent>
    <extension base="tns:RespType">
      <attribute name="timeStamp" type="long integer"/>
    </extension>
  </complexContent>
</complexType>

<complexType name="PingReqType">
  <complexContent>
    <extension base="tns:ReqType">
      <attribute name="timeStamp" type="long integer"/>
    </extension>
  </complexContent>
</complexType>

<!-- ===== Protocol Elements ===== -->

<element name="CreateAccount" type="tns:CreateAccountType"/>
<element name="DeleteAccount" type="tns:DeleteAccountType"/>
<element name="GetAccount" type="tns:GetAccountType"/>
<element name="SetAccount" type="tns:SetAccountType"/>
<element name="EventAccountCnf" type="tns:AccountType"/>
<element name="CreateAccountResp" type="tns:CreateGetSetAccountRespType"/>
<element name="DeleteAccountResp" type="tns:RespType"/>
<element name="GetAccountResp" type="tns:CreateGetSetAccountRespType"/>
<element name="SetAccountResp" type="tns:CreateGetSetAccountRespType"/>

<element name="GetAccountSummary" type="tns:ReqType"/>
<element name="GetAccountSummaryResp" type="tns:GetSummaryRespType"/>
<element name="EventAccountSummary" type="tns:EventAccountSummaryType"/>

<element name="EventACLConf" type="tns:EventACLConfType"/>
<element name="CreateACLEntry" type="tns:CreateSetACLEntryType"/>
<element name="DeleteACLEntry" type="tns:DeleteACLEntryType"/>
<element name="GetACLEntry" type="tns:GetACLEntryType"/>
<element name="SetACLEntry" type="tns:CreateSetACLEntryType"/>
<element name="CreateACLEntryResp" type="tns:CreateGetSetACLEntryRespType"/>
<element name="DeleteACLEntryResp" type="tns:RespType"/>
<element name="GetACLEntryResp" type="tns:CreateGetSetACLEntryRespType"/>
<element name="SetACLEntryResp" type="tns:CreateGetSetACLEntryRespType"/>

<element name="GetActivePPDev" type="tns:GetActivePPDevType"/>
<element name="GetActivePPDevResp" type="tns:GetActivePPDevRespType"/>

<element name="GetAdvancedSIP" type="tns:ReqType"/>
<element name="SetAdvancedSIP" type="tns:SetAdvancedSIPType"/>
<element name="EventAdvancedSIPCnf" type="tns:AdvancedSIPType"/>
<element name="GetAdvancedSIPResp" type="tns:GetSetAdvancedSIPRespType"/>
<element name="SetAdvancedSIPResp" type="tns:GetSetAdvancedSIPRespType"/>

```

```

<element name="GetConferenceServerSIP" type="tns:ReqType"/>
<element name="SetConferenceServerSIP" type="tns:SetConferenceServerSIPType"/>
<element name="EventConferenceServerSIPCnf" type="tns:ConferenceServerSIPType"/>
<element name="GetConferenceServerSIPResp" type="tns:GetSetConferenceServerSIPRespType"/>
<element name="SetConferenceServerSIPResp" type="tns:GetSetConferenceServerSIPRespType"/>

<element name="GetPortRangeSIP" type="tns:ReqType"/>
<element name="SetPortRangeSIP" type="tns:SetPortRangeSIPType"/>
<element name="EventPortRangeSIPCnf" type="tns:PortRangeSIPType"/>
<element name="GetPortRangeSIPResp" type="tns:GetSetPortRangeSIPRespType"/>
<element name="SetPortRangeSIPResp" type="tns:GetSetPortRangeSIPRespType"/>

<element name="GetIntercomCallHandling" type="tns:ReqType"/>
<element name="SetIntercomCallHandling" type="tns:SetIntercomCallHandlingType"/>
<element name="EventIntercomCallHandlingCnf" type="tns:IntercomCallHandlingType"/>
<element name="GetIntercomCallHandlingResp" type="tns:GetSetIntercomCallHandlingRespType"/>
<element name="SetIntercomCallHandlingResp" type="tns:GetSetIntercomCallHandlingRespType"/>

<element name="GetBackupSIP" type="tns:ReqType"/>
<element name="SetBackupSIP" type="tns:SetBackupSIPType"/>
<element name="EventBackupSIPCnf" type="tns:BackupSIPType"/>
<element name="GetBackupSIPResp" type="tns:GetSetBackupSIPRespType"/>
<element name="SetBackupSIPResp" type="tns:GetSetBackupSIPRespType"/>

<element name="EventAlarmCallProgress" type="tns:EventAlarmCallProgressType"/>

<element name="CreateAlarmTrigger" type="tns:CreateAlarmTriggerType"/>
<element name="DeleteAlarmTrigger" type="tns>DeleteAlarmTriggerType"/>
<element name="EventAlarmTrigger" type="tns:EventAlarmTriggerType"/>
<element name="GetAlarmTrigger" type="tns:GetAlarmTriggerType"/>
<element name="SetAlarmTrigger" type="tns:SetAlarmTriggerType"/>
<element name="EventAlarmTriggerCnf" type="tns:EventAlarmTriggerCnfType"/>
<element name="CreateAlarmTriggerResp" type="tns>CreateGetSetAlarmTriggerRespType"/>
<element name="DeleteAlarmTriggerResp" type="tns:RespType"/>
<element name="GetAlarmTriggerResp" type="tns>CreateGetSetAlarmTriggerRespType"/>
<element name="SetAlarmTriggerResp" type="tns>CreateGetSetAlarmTriggerRespType"/>

<element name="GetAlarmTriggerSummary" type="tns:ReqType"/>
<element name="GetAlarmTriggerSummaryResp" type="tns:GetSummaryRespType"/>

<element name="GetAutoDBBackup" type="tns:ReqType"/>
<element name="SetAutoDBBackup" type="tns:SetAutoDBBackupType"/>
<element name="EventAutoDBBackupCnf" type="tns:AutoDBBackupType"/>
<element name="GetAutoDBBackupResp" type="tns:GetSetAutoDBBackupRespType"/>
<element name="SetAutoDBBackupResp" type="tns:GetSetAutoDBBackupRespType"/>
<element name="GetAutoDBBackupFileName" type="tns:ReqType"/>
<element name="EventAutoDBBackupFileNameCnf" type="tns:AutoDBBackupFilenameType"/>
<element name="GetAutoDBBackupFileNameResp" type="tns:GetAutoDBBackupFilenameRespType"/>

<element name="EventDbTransferState" type="tns:EventDbTransferStateType"/>
<element name="GetDbTransferState" type="tns:GetDbTransferStateType"/>
<element name="GetDbTransferStateResp" type="tns:GetDbTransferStateRespType"/>
<element name="SetDbTransferState" type="tns:SetDbTransferStateType"/>
<element name="SetDbTransferStateResp" type="tns:SetDbTransferStateRespType"/>

<element name="GetBasicSIP" type="tns:ReqType"/>
<element name="SetBasicSIP" type="tns:SetBasicSIPType"/>
<element name="EventBasicSIPCnf" type="tns:BasicSIPType"/>
<element name="GetBasicSIPResp" type="tns:GetSetBasicSIPRespType"/>
<element name="SetBasicSIPResp" type="tns:GetSetBasicSIPRespType"/>

```

```

<element name="GetSecureSIP" type="tns:ReqType"/>
<element name="SetSecureSIP" type="tns:SetSecureSIPType"/>
<element name="EventSecureSIPCnf" type="tns:SecureSIPType"/>
<element name="GetSecureSIPResp" type="tns:GetSetSecureSIPRespType"/>
<element name="SetSecureSIPResp" type="tns:GetSetSecureSIPRespType"/>

<element name="SetSecureOMMCertificate" type="tns:SetSecureOMMCertificateType"/>
<element name="GetSecureOMMCertificate" type="tns:ReqType"/>
<element name="SetSecureOMMCertificateResp" type="tns:GetSetSecureOMMCertificateRespType"/>
<element name="GetSecureOMMCertificateResp" type="tns:GetSetSecureOMMCertificateRespType"/>
<element name="SetSecureOMMCertificateServerImport" type="tns:SetSecureOMMCertificateServerImportType"/>
<element name="GetSecureOMMCertificateServerImport" type="tns:ReqType"/>
<element name="EventSecureOMMCertificateServerImportCnf" type="tns:SecureOMMCertificateServerImportType"/>
<element name="SetSecureOMMCertificateServerImportResp" type="tns:GetSetSecureOMMCertificateServerImportRespType"/>
<element name="GetSecureOMMCertificateServerImportResp" type="tns:GetSetSecureOMMCertificateServerImportRespType"/>

<element name="SetSecurePROVCertificateServerImport" type="tns:SetSecurePROVCertificateServerImportType"/>
<element name="GetSecurePROVCertificateServerImport" type="tns:ReqType"/>
<element name="EventSecurePROVCertificateServerImportCnf" type="tns:SecurePROVCertificateServerImportType"/>
<element name="SetSecurePROVCertificateServerImportResp" type="tns:GetSetSecurePROVCertificateServerImportRespType"/>
<element name="GetSecurePROVCertificateServerImportResp" type="tns:GetSetSecurePROVCertificateServerImportRespType"/>

<element name="SetSecureSIPCertificate" type="tns:SetSecureSIPCertificateType"/>
<element name="GetSecureSIPCertificate" type="tns:ReqType"/>
<element name="EventSecureSIPCertificateCnf" type="tns:EventSecureSIPCertificateCnfType"/>
<element name="SetSecureSIPCertificateResp" type="tns:GetSetSecureSIPCertificateRespType"/>
<element name="GetSecureSIPCertificateResp" type="tns:GetSetSecureSIPCertificateRespType"/>
<element name="SetSecureSIPCertificateServerImport" type="tns:SetSecureSIPCertificateServerImportType"/>
<element name="GetSecureSIPCertificateServerImport" type="tns:ReqType"/>
<element name="EventSecureSIPCertificateServerImportCnf" type="tns:SecureSIPCertificateServerImportType"/>
<element name="SetSecureSIPCertificateServerImportResp" type="tns:GetSetSecureSIPCertificateServerImportRespType"/>
<element name="GetSecureSIPCertificateServerImportResp" type="tns:GetSetSecureSIPCertificateServerImportRespType"/>

<element name="SetOMMCertificate" type="tns:SetOMMCertificateType"/>
<element name="GetOMMCertificate" type="tns:ReqType"/>
<element name="EventOMMCertificateCnf" type="tns:EventOMMCertificateCnfType"/>
<element name="SetOMMCertificateResp" type="tns:GetSetOMMCertificateRespType"/>
<element name="GetOMMCertificateResp" type="tns:GetSetOMMCertificateRespType"/>

<element name="GetDECTAuthCode" type="tns:ReqType"/>
<element name="SetDECTAuthCode" type="tns:SetDECTAuthCodeType"/>
<element name="EventDECTAuthCodeCnf" type="tns:DECTAuthCodeType"/>
<element name="GetDECTAuthCodeResp" type="tns:GetSetDECTAuthCodeRespType"/>
<element name="SetDECTAuthCodeResp" type="tns:GetSetDECTAuthCodeRespType"/>

<element name="GetDECTEncryption" type="tns:ReqType"/>
<element name="SetDECTEncryption" type="tns:SetDECTEncryptionType"/>
<element name="EventDECTEncryptionCnf" type="tns:DECTEncryptionType"/>
<element name="GetDECTEncryptionResp" type="tns:GetSetDECTEncryptionRespType"/>
<element name="SetDECTEncryptionResp" type="tns:GetSetDECTEncryptionRespType"/>

<element name="GetDECTPagingAreaSize" type="tns:ReqType"/>
<element name="SetDECTPagingAreaSize" type="tns:SetDECTPagingAreaSizeType"/>
<element name="EventDECTPagingAreaSizeCnf" type="tns:DECTPagingAreaSizeType"/>
<element name="GetDECTPagingAreaSizeResp" type="tns:GetSetDECTPagingAreaSizeRespType"/>
<element name="SetDECTPagingAreaSizeResp" type="tns:GetSetDECTPagingAreaSizeRespType"/>

<element name="GetDECTRegDomain" type="tns:ReqType"/>
<element name="SetDECTRegDomain" type="tns:SetDECTRegDomainType"/>
<element name="EventDECTRegDomainCnf" type="tns:DECTRegDomainType"/>
<element name="GetDECTRegDomainResp" type="tns:GetSetDECTRegDomainRespType"/>
<element name="SetDECTRegDomainResp" type="tns:GetSetDECTRegDomainRespType"/>

```

```

<element name="GetDECTSubscriptionMode" type="tns:ReqType"/>
<element name="EventDECTSubscriptionMode" type="tns:DECTSubscriptionModeType"/>
<element name="SetDECTSubscriptionMode" type="tns:SetDECTSubscriptionModeType"/>
<element name="GetDECTSubscriptionModeResp" type="tns:GetSetDECTSubscriptionModeType"/>
<element name="SetDECTSubscriptionModeResp" type="tns:GetSetDECTSubscriptionModeType"/>

<element name="GetDECTPpSettings" type="tns:ReqType"/>
<element name="SetDECTPpSettings" type="tns:SetDECTPpSettingsType"/>
<element name="EventDECTPpSettingsCnf" type="tns:DECTPpSettingsType"/>
<element name="GetDECTPpSettingsResp" type="tns:GetSetDECTPpSettingsRespType"/>
<element name="SetDECTPpSettingsResp" type="tns:GetSetDECTPpSettingsRespType"/>

<element name="GetDevAutoCreate" type="tns:ReqType"/>
<element name="SetDevAutoCreate" type="tns:SetDevAutoCreateType"/>
<element name="EventDevAutoCreateCnf" type="tns:DevAutoCreateType"/>
<element name="GetDevAutoCreateResp" type="tns:GetSetDevAutoCreateRespType"/>
<element name="SetDevAutoCreateResp" type="tns:GetSetDevAutoCreateRespType"/>

<element name="GetRestrictedSubscriptionDuration" type="tns:ReqType"/>
<element name="SetRestrictedSubscriptionDuration" type="tns:SetRestrictedSubscriptionDurationType"/>
<element name="EventRestrictedSubscriptionDurationCnf" type="tns:RestrictedSubscriptionDurationType"/>
<element name="GetRestrictedSubscriptionDurationResp" type="tns:GetSetRestrictedSubscriptionDurationRespType"/>
<element name="SetRestrictedSubscriptionDurationResp" type="tns:GetSetRestrictedSubscriptionDurationRespType"/>

<element name="CreateDigitTreatment" type="tns:CreateSetDigitTreatmentType"/>
<element name="DeleteDigitTreatment" type="tns>DeleteDigitTreatmentType"/>
<element name="GetDigitTreatment" type="tns:GetDigitTreatmentType"/>
<element name="SetDigitTreatment" type="tns:CreateSetDigitTreatmentType"/>
<element name="EventDigitTreatmentCnf" type="tns:EventDigitTreatmentType"/>
<element name="CreateDigitTreatmentResp" type="tns:CreateSetDigitTreatmentRespType"/>
<element name="DeleteDigitTreatmentResp" type="tns:RespType"/>
<element name="GetDigitTreatmentResp" type="tns:GetDigitTreatmentRespType"/>
<element name="SetDigitTreatmentResp" type="tns:CreateSetDigitTreatmentRespType"/>

<element name="GetDigitTreatmentSummary" type="tns:ReqType"/>
<element name="GetDigitTreatmentSummaryResp" type="tns:GetSummaryRespType"/>

<element name="GetDTMF" type="tns:ReqType"/>
<element name="SetDTMF" type="tns:SetDTMFType"/>
<element name="EventDTMFCnf" type="tns:DTMFType"/>
<element name="GetDTMFResp" type="tns:GetSetDTMFRespType"/>
<element name="SetDTMFResp" type="tns:GetSetDTMFRespType"/>

<element name="GetEULAConfirm" type="tns:ReqType"/>
<element name="SetEULAConfirm" type="tns:SetEULAConfirmType"/>
<element name="EventEULAConfirmCnf" type="tns:EULAConfirmType"/>
<element name="GetEULAConfirmResp" type="tns:GetSetEULAConfirmRespType"/>
<element name="SetEULAConfirmResp" type="tns:GetSetEULAConfirmRespType"/>

<element name="DeleteEventLogBuffer" type="tns:ReqType"/>
<element name="GetEventLogBuffer" type="tns:GetEventLogBufferType"/>
<element name="DeleteEventLogBufferResp" type="tns:RespType"/>
<element name="GetEventLogBufferResp" type="tns:GetEventLogBufferRespType"/>
<element name="EventEventLogBuffer" type="tns:EventLogBufferEntryType"/>

<element name="GetFACList" type="tns:ReqType"/>
<element name="SetFAC" type="tns:SetFACType"/>
<element name="SetFACList" type="tns:SetFACListType"/>
<element name="EventFACCnf" type="tns:FACSequenceType"/>
<element name="GetFACListResp" type="tns:GetFACListRespType"/>
<element name="SetFACResp" type="tns:SetFACListRespType"/>

```



```

<element name="SetFACListResp" type="tns:SetFACListRespType"/>
<element name="GetFACPrefix" type="tns:ReqType"/>
<element name="SetFACPrefix" type="tns:SetFACPrefixType"/>
<element name="EventFACPrefixCnf" type="tns:FACPrefixType"/>
<element name="GetFACPrefixResp" type="tns:GetSetFACPrefixRespType"/>
<element name="SetFACPrefixResp" type="tns:GetSetFACPrefixRespType"/>

<element name="GetFile" type="tns:GetFileType"/>
<element name="GetFileResp" type="tns:GetFileRespType"/>

<element name="CreateFixedPP" type="tns:CreateFixedPPType"/>
<element name="CreateFixedPPResp" type="tns:CreateFixedPPRespType"/>

<element name="GetFlashMemUsage" type="tns:ReqType"/>
<element name="GetFlashMemUsageResp" type="tns:EventFlashMemUsageRespType"/>

<element name="EventHealthState" type="tns:HealthType"/>
<element name="GetHealthState" type="tns:ReqType"/>
<element name="GetHealthStateResp" type="tns:GetHealthStateRespType"/>

<element name="GetIMA" type="tns:ReqType"/>
<element name="SetIMA" type="tns:SetIMAType"/>
<element name="EventIMACnf" type="tns:ServerType"/>
<element name="GetIMARes" type="tns:GetSetIMAResType"/>
<element name="SetIMARes" type="tns:GetSetIMAResType"/>
<element name="UpdateIMAConfigFile" type="tns:ReqType"/>
<element name="UpdateIMAConfigFileResp" type="tns:RespType"/>

<element name="GetOMPURL" type="tns:ReqType"/>
<element name="SetOMPURL" type="tns:SetOMPURLType"/>
<element name="EventOMPURLCnf" type="tns:ServerType"/>
<element name="GetOMPURLResp" type="tns:GetSetOMPURLRespType"/>
<element name="SetOMPURLResp" type="tns:GetSetOMPURLRespType"/>

<element name="GetLastPPDevAction" type="tns:GetRecordType"/>
<element name="GetLastPPDevActionResp" type="tns:GetActivePPDevRespType"/>

<element name="CreateLDAP" type="tns:CreateSetLDAPType"/>
<element name="DeleteLDAP" type="tns>DeleteLDAPType"/>
<element name="GetLDAP" type="tns:GetLDAPType"/>
<element name="SetLDAP" type="tns:CreateSetLDAPType"/>
<element name="EventLDAPCnf" type="tns:EventLDAPType"/>
<element name="SetLDAPResp" type="tns:CreateSetLDAPRespType"/>
<element name="CreateLDAPResp" type="tns:CreateSetLDAPRespType"/>
<element name="DeleteLDAPResp" type="tns:RespType"/>
<element name="GetLDAPResp" type="tns:GetLDAPRespType"/>

<element name="GetLicense" type="tns:ReqType"/>
<element name="EventLicenseCnf" type="tns:LicenseType"/>
<element name="EventLicensedCodecLines" type="tns:LicensedCodecLinesType"/>
<element name="GetLicensedCodecLines" type="tns:ReqType"/>
<element name="GetLicensedCodecLinesResp" type="tns:LicensedCodecLinesRespType"/>
<element name="GetLicenseResp" type="tns:LicenseRespType"/>

<element name="GetLicenseServerList" type="tns:ReqType"/>
<element name="SetLicenseServerList" type="tns:SetLicenseServerListType"/>
<element name="EventLicenseServerListCnf" type="tns:LicenseServerListType"/>
<element name="GetLicenseResp" type="tns:GetSetLicenseServerListRespType"/>
<element name="SetLicenseResp" type="tns:GetSetLicenseServerListRespType"/>
<element name="EventLicenseFile" type="tns:LicenseFileType"/>

<element name="DeleteMessage" type="tns:CancelDeleteMessageType"/>

```

```

<element name="EventMessageConfirmation" type="tns:EventMessageConfirmationType"/>
<element name="EventMessageProgress" type="tns:EventMessageProgressType"/>
<element name="EventMessageQueueEmpty" type="tns:EventMessageQueueEmptyType"/>
<element name="DeleteMessageResp" type="tns:RespType"/>
<element name="EventMessageSend" type="tns:MessageSendType"/>
<element name="CancelMessageResp" type="tns:RespType"/>
<element name="CancelMessage" type="tns:CancelDeleteMessageType"/>
<element name="SendMessageResp" type="tns:RespType"/>
<element name="SendMessage" type="tns:SendMessageType"/>

<element name="GetNetParams" type="tns:ReqType"/>
<element name="SetNetParams" type="tns:SetNetParamsType"/>
<element name="EventNetParamsCnf" type="tns:NetParamsType"/>
<element name="GetNetParamsResp" type="tns:GetSetNetParamsRespType"/>
<element name="SetNetParamsResp" type="tns:GetSetNetParamsRespType"/>

<element name="GetDECTPowerLimit" type="tns:ReqType"/>
<element name="SetDECTPowerLimit" type="tns:SetDECTPowerLimitType"/>
<element name="EventDECTPowerLimitCnf" type="tns:DECTPowerLimitType"/>
<element name="GetDECTPowerLimitResp" type="tns:GetSetDECTPowerLimitRespType"/>
<element name="SetDECTPowerLimitResp" type="tns:GetSetDECTPowerLimitRespType"/>

<element name="SetPARK" type="tns:SetPARKType"/>
<element name="GetPARK" type="tns:ReqType"/>
<element name="EventPARKCnf" type="tns:PARKType"/>
<element name="GetPARKResp" type="tns:GetSetPARKRespType"/>
<element name="SetPARKResp" type="tns:GetSetPARKRespType"/>
<element name="EventPARKCnf" type="tns:PARKType"/>
<element name="PARKFromServer" type="tns:ReqType"/>
<element name="EventPARKFromServerResult" type="tns:PARKFromServerResultType"/>

<element name="SetSARI" type="tns:SetSARIType"/>
<element name="GetSARI" type="tns:ReqType"/>
<element name="EventSARICnf" type="tns:SARIType"/>
<element name="GetSARIResp" type="tns:GetSetSARIRespType"/>
<element name="SetSARIResp" type="tns:GetSetSARIRespType"/>
<element name="EventSARICnf" type="tns:SARIType"/>

<element name="GetPBXEnrolment" type="tns:ReqType"/>
<element name="SetPBXEnrolment" type="tns:SetPBXEnrolmentType"/>
<element name="EventPBXEnrolmentCnf" type="tns:PBXEnrolmentType"/>
<element name="GetPBXEnrolmentResp" type="tns:GetSetPBXEnrolmentRespType"/>
<element name="SetPBXEnrolmentResp" type="tns:GetSetPBXEnrolmentRespType"/>

<element name="EventPermissionChange" type="tns:PermissionChangeEvent"/>

<element name="EventPositionHistory" type="tns:EventPositionHistoryType"/>
<element name="EventPositionInfo" type="tns:EventPositionInfoType"/>
<element name="EventPositionRequest" type="tns:EventPositionRequestType"/>
<element name="EventPositionTrack" type="tns:EventPositionTrackType"/>

<element name="CreatePPDev" type="tns:CreatePPDevType"/>
<element name="DeletePPDev" type="tns>DeletePPDevType"/>
<element name="GetPPDev" type="tns:ReqType"/>
<element name="SetPPDev" type="tns:SetPPDevType"/>
<element name="EventPPDevCnf" type="tns:EventPPDevType"/>
<element name="CreatePPDevResp" type="tns>CreateGetSetPPDevRespType"/>
<element name="GetPPDevResp" type="tns>CreateGetSetPPDevRespType"/>
<element name="DeletePPDevResp" type="tns:RespType"/>
<element name="SetPPDevResp" type="tns>CreateGetSetPPDevRespType"/>

<element name="EventPPDevSummary" type="tns:PPDevSummaryType"/>

```

```

<element name="GetPPDevSummary" type="tns:ReqType"/>
<element name="GetPPDevSummaryResp" type="tns:GetPPDevSummaryRespType"/>

<element name="GetPPFirmwareUpdate" type="tns:ReqType"/>
<element name="SetPPFirmwareUpdate" type="tns:SetPPFirmwareUpdateType"/>
<element name="EventPPFirmwareUpdateCnf" type="tns:PPFirmwareUpdateType"/>
<element name="GetPPFirmwareUpdateResp" type="tns:GetSetPPFirmwareUpdateRespType"/>
<element name="SetPPFirmwareUpdateResp" type="tns:GetSetPPFirmwareUpdateRespType"/>

<element name="EventPPFirmwareUpdateOverview" type="tns:PPFirmwareUpdateOverviewType"/>
<element name="GetPPFirmwareUpdateOverview" type="tns:ReqType"/>
<element name="GetPPFirmwareUpdateOverviewResp" type="tns:GetPPFirmwareUpdateOverviewRespType"/>

<element name="EventPPFirmwareUpdateStatus" type="tns:EventPPFirmwareUpdateStatusType"/>
<element name="GetPPFirmwareUpdateStatus" type="tns:GetPPFirmwareUpdateStatusType"/>
<element name="GetPPFirmwareUpdateStatusResp" type="tns:GetPPFirmwareUpdateStatusRespType"/>

<element name="GetPPLoginVariant" type="tns:ReqType"/>
<element name="SetPPLoginVariant" type="tns:SetPPLoginVariantType"/>
<element name="EventPPLoginVariantCnf" type="tns:PPLoginVariantType"/>
<element name="GetPPLoginVariantResp" type="tns:GetSetPPLoginVariantRespType"/>
<element name="SetPPLoginVariantResp" type="tns:GetSetPPLoginVariantRespType"/>

<element name="EventPPState" type="tns:PPStateType"/>
<element name="GetPPState" type="tns:GetPPStateType"/>
<element name="GetPPStateResp" type="tns:GetPPStateRespType"/>

<element name="EventPPTransaction" type="tns:EventPPTransactionType"/>

<element name="CreatePPUser" type="tns>CreateSetPPUserType"/>
<element name="DeletePPUser" type="tns>DeletePPUserType"/>
<element name="GetPPUser" type="tns:GetPPUserType"/>
<element name="SetPPUser" type="tns>CreateSetPPUserType"/>
<element name="EventPPUserCnf" type="tns:EventPPUserType"/>
<element name="CreatePPUserResp" type="tns>CreateSetPPUserRespType"/>
<element name="DeletePPUserResp" type="tns:RespType"/>
<element name="GetPPUserResp" type="tns:GetPPUserRespType"/>
<element name="SetPPUserResp" type="tns>CreateSetPPUserRespType"/>

<element name="SetPPUserDevRelation" type="tns:SetPPUserDevRelationType"/>
<element name="SetPPUserDevRelationResp" type="tns:SetPPUserDevRelationRespType"/>

<element name="EventPPUserSummary" type="tns:PPUserSummaryType"/>
<element name="GetPPUserSummary" type="tns:ReqType"/>
<element name="GetPPUserSummaryResp" type="tns:GetPPUserSummaryRespType"/>

<element name="SetPPUserTracking" type="tns:SetPPUserTrackingType"/>
<element name="SetPPUserTrackingResp" type="tns:SetPPUserTrackingRespType"/>

<element name="GetPublicKey" type="tns:ReqType"/>
<element name="GetPublicKeyResp" type="tns:GetPublicKeyRespType"/>

<element name="GetRegistrationTrafficShaping" type="tns:ReqType"/>
<element name="SetRegistrationTrafficShaping" type="tns:SetRegistrationTrafficShapingType"/>
<element name="EventRegistrationTrafficShapingCnf" type="tns:RegistrationTrafficShapingType"/>
<element name="GetRegistrationTrafficShapingResp" type="tns:GetSetRegistrationTrafficShapingRespType"/>
<element name="SetRegistrationTrafficShapingResp" type="tns:GetSetRegistrationTrafficShapingRespType"/>

<element name="GetRemoteAccess" type="tns:ReqType"/>
<element name="SetRemoteAccess" type="tns:SetRemoteAccessType"/>
<element name="EventRemoteAccessCnf" type="tns:RemoteAccess"/>
<element name="GetRemoteAccessResp" type="tns:GetSetRemoteAccessRespType"/>

```

```

<element name="SetRemoteAccessResp" type="tns:GetSetRemoteAccessRespType"/>

<element name="CreateRFP" type="tns:CreateSetRFPTType"/>
<element name="DeleteRFP" type="tns:DeleteGetRFPTType"/>
<element name="GetRFP" type="tns:DeleteGetRFPTType"/>
<element name="SetRFP" type="tns:CreateSetRFPTType"/>
<element name="EventRFPConf" type="tns:RFPTType"/>
<element name="GetRFPResp" type="tns:CreateGetSetRFPRespType"/>
<element name="SetRFPResp" type="tns:CreateGetSetRFPRespType"/>
<element name="CreateRFPResp" type="tns:CreateRFPRespType"/>
<element name="GetRFPResp" type="tns:CreateGetSetRFPRespType"/>
<element name="DeleteRFPResp" type="tns:RespType"/>

<element name="GetRFPFM" type="tns:GetRFPFMType"/>
<element name="SetRFPFM" type="tns:SetRFPFMType"/>
<element name="EventRFPFMConf" type="tns:RFPFMType"/>
<element name="SetRFPFMResp" type="tns:GetSetRFPFMRespType"/>
<element name="GetRFPFMResp" type="tns:CreateGetSetRFPFMRespType"/>

<element name="EventRFPCapture" type="tns:EventRFPCaptureType"/>
<element name="EventRFPCaptureConf" type="tns:RFPCaptureType"/>
<element name="GetRFPCapture" type="tns:ReqType"/>
<element name="SetRFPCapture" type="tns:SetRFPCaptureType"/>
<element name="GetRFPCaptureResp" type="tns:GetSetRFPCaptureRespType"/>
<element name="SetRFPCaptureResp" type="tns:GetSetRFPCaptureRespType"/>

<element name="DeleteRFPCaptureList" type="tns:ReqType"/>
<element name="DeleteRFPCaptureListElem" type="tns:DeleteRFPCaptureListElemType"/>
<element name="DeleteRFPCaptureListElemResp" type="tns:RespType"/>
<element name="DeleteRFPCaptureListResp" type="tns:RespType"/>
<element name="GetRFPCaptureList" type="tns:ReqType"/>
<element name="GetRFPCaptureListResp" type="tns:GetRFPCaptureListRespType"/>
<element name="EventRFPCaptureAttempt" type="tns:RFPCaptureAttemptType"/>

<element name="EventRFPDetails" type="tns:EventRFPType"/>

<element name="EventRFPState" type="tns:EventRFPType"/>

<element name="GetRFPStatistic" type="tns:GetRFPStatisticType"/>
<element name="GetRFPStatisticConfig" type="tns:ReqType"/>
<element name="GetRFPStatisticConfigResp" type="tns:GetRFPStatisticConfigRespType"/>
<element name="GetRFPStatisticResp" type="tns:GetRFPStatisticRespType"/>

<element name="EventRFPSummary" type="tns:RFPSummaryType"/>
<element name="GetRFPSummary" type="tns:ReqType"/>
<element name="GetRFPSummaryResp" type="tns:GetRFPSummaryRespType"/>

<element name="GetRFPSSync" type="tns:GetRFPSSyncType"/>
<element name="EventRFPSSyncRel" type="tns:EventRFPSSyncRelType"/>
<element name="GetRFPSSyncResp" type="tns:GetSetRFPSSyncRespType"/>

<element name="EventRFPQuality" type="tns:EventRFPQualityType"/>
<element name="GetRFPQuality" type="tns:GetRFPQualityType"/>
<element name="GetRFPQualityResp" type="tns:GetRFPQualityRespType"/>
<element name="EventRFPMediaStreamQuality" type="tns:EventRFPMediaStreamQualityType"/>
<element name="GetRFPMediaStreamQuality" type="tns:GetRFPMediaStreamQualityType"/>
<element name="GetRFPMediaStreamQualityResp" type="tns:GetRFPMediaStreamQualityRespType"/>
<element name="ResetRFPMediaStreamQuality" type="tns:ResetRFPMediaStreamQualityType"/>
<element name="EventRFPSSyncQuality" type="tns:EventRFPSSyncQualityType"/>
<element name="GetRFPSSyncQuality" type="tns:GetRFPSSyncQualityType"/>
<element name="GetRFPSSyncQualityResp" type="tns:GetRFPSSyncQualityRespType"/>

```

```

<element name="GetRTP" type="tns:ReqType" />
<element name="SetRTP" type="tns:SetRTPType" />
<element name="EventRTPCnf" type="tns:RTPType" />
<element name="GetRTPResp" type="tns:GetSetRTPRespType" />
<element name="SetRTPResp" type="tns:GetSetRTPRespType" />

<element name="CreateSite" type="tns:CreateSetSiteType" />
<element name="DeleteSite" type="tns>DeleteSiteType" />
<element name="GetSite" type="tns:GetRecordType" />
<element name="SetSite" type="tns:CreateSetSiteType" />
<element name="EventSiteCnf" type="tns:EventSiteType" />
<element name="CreateSiteResp" type="tns:CreateSetSiteRespType" />
<element name="DeleteSiteResp" type="tns:RespType" />
<element name="GetSiteResp" type="tns:GetSiteRespType" />
<element name="SetSiteResp" type="tns:CreateSetSiteRespType" />
<element name="GetSiteSummary" type="tns:ReqType" />
<element name="GetSiteSummaryResp" type="tns:GetSiteSummaryRespType" />
<element name="EventSiteSummary" type="tns:EventSiteSummaryType" />

<element name="GetSNMP" type="tns:ReqType" />
<element name="SetSNMP" type="tns:SetSNMPType" />
<element name="EventSNMPCnf" type="tns:SNMPType" />
<element name="GetSNMPResp" type="tns:GetSetSNMPRespType" />
<element name="SetSNMPResp" type="tns:GetSetSNMPRespType" />

<element name="GetSpecialBranding" type="tns:ReqType" />
<element name="SetSpecialBranding" type="tns:SetSpecialBrandingType" />
<element name="EventSpecialBrandingCnf" type="tns:SpecialBrandingType" />
<element name="GetSpecialBrandingResp" type="tns:GetSetSpecialBrandingRespType" />
<element name="SetSpecialBrandingResp" type="tns:GetSetSpecialBrandingRespType" />

<element name="GetSoftwareUpdate" type="tns:ReqType" />
<element name="EventSoftwareUpdateCnf" type="tns:SoftwareUpdateType" />
<element name="GetSoftwareUpdateResp" type="tns:SoftwareUpdateRespType" />
<element name="SoftwareUpdateResp" type="tns:SoftwareUpdateRespType" />
<element name="SoftwareUpdate" type="tns:SoftwareUpdateReqType" />

<element name="GetStbState" type="tns:ReqType" />
<element name="EventStbStateChange" type="tns:StbStateType" />
<element name="GetStbStateResp" type="tns:StbStateRespType" />

<element name="GetSuplServ" type="tns:ReqType" />
<element name="SetSuplServ" type="tns:SetSuplServType" />
<element name="EventSuplServCnf" type="tns:SuplServType" />
<element name="GetSuplServResp" type="tns:GetSetSuplServRespType" />
<element name="SetSuplServResp" type="tns:GetSetSuplServRespType" />

<element name="GetSyslogServer" type="tns:ReqType" />
<element name="SetSyslogServer" type="tns:SetSyslogServerType" />
<element name="EventSyslogServerCnf" type="tns:SyslogServerType" />
<element name="GetSyslogServerResp" type="tns:GetSetSyslogServerRespType" />
<element name="SetSyslogServerResp" type="tns:GetSetSyslogServerRespType" />

<element name="ActivateRemoteSystemDump" type="tns:ReqType" />
<element name="ActivateRemoteSystemDumpResp" type="tns:RespType" />
<element name="GetRemoteSystemDump" type="tns:ReqType" />
<element name="SetRemoteSystemDump" type="tns:SetRemoteSystemDumpType" />
<element name="EventRemoteSystemDumpCnf" type="tns:RemoteSystemDumpType" />
<element name="GetRemoteSystemDumpResp" type="tns:GetSetRemoteSystemDumpRespType" />
<element name="SetRemoteSystemDumpResp" type="tns:GetSetRemoteSystemDumpRespType" />

<element name="GetSysStatisticConfig" type="tns:ReqType" />

```

```

<element name="GetSysStatisticConfigResp" type="tns:GetSysStatisticConfigConfigRespType"/>
<element name="GetSysStatisticMinMax" type="tns:GetSysStatisticMinMaxType"/>
<element name="GetSysStatisticMinMaxRecord" type="tns:GetSysStatisticMinMaxRecordType"/>
<element name="GetSysStatisticMinMaxRecordResp" type="tns:GetSysStatisticMinMaxRecordRespType"/>
<element name="GetSysStatisticMinMaxResp" type="tns:GetSysStatisticMinMaxRespType"/>
<element name="GetSysStatisticMinMaxSummary" type="tns:GetSysStatisticMinMaxSummaryType"/>
<element name="GetSysStatisticMinMaxSummaryResp" type="tns:GetSysStatisticMinMaxSummaryRespType"/>
<element name="GetSysStatisticOccurrence" type="tns:GetSysStatisticOccurrenceType"/>
<element name="GetSysStatisticOccurrenceResp" type="tns:GetSysStatisticOccurrenceRespType"/>

<element name="GetSystemName" type="tns:ReqType"/>
<element name="SetSystemName" type="tns:SetSystemNameType"/>
<element name="EventSystemNameCnf" type="tns:SystemNameType"/>
<element name="GetSystemNameResp" type="tns:GetSetSystemNameRespType"/>
<element name="SetSystemNameResp" type="tns:GetSetSystemNameRespType"/>

<element name="GetSystemCredentials" type="tns:ReqType"/>
<element name="SetSystemCredentials" type="tns:SetSystemCredentialsType"/>
<element name="EventSystemCredentialsCnf" type="tns:SystemCredentialsType"/>
<element name="GetSystemCredentialsResp" type="tns:GetSetSystemCredentialsRespType"/>
<element name="SetSystemCredentialsResp" type="tns:GetSetSystemCredentialsRespType"/>

<element name="GetSystemProvUpdTrig" type="tns:ReqType"/>
<element name="SetSystemProvUpdTrig" type="tns:SetSystemProvUpdTrigType"/>
<element name="EventSystemProvUpdTrigCnf" type="tns:SystemProvUpdTrigType"/>
<element name="GetSystemProvUpdTrigResp" type="tns:GetSetSystemProvUpdTrigRespType"/>
<element name="SetSystemProvUpdTrigResp" type="tns:GetSetSystemProvUpdTrigRespType"/>

<element name="GetConfigURL" type="tns:ReqType"/>
<element name="SetConfigURL" type="tns:SetConfigURLType"/>
<element name="EventConfigURLCnf" type="tns:ConfigURLType"/>
<element name="GetConfigURLResp" type="tns:GetSetConfigURLRespType"/>
<element name="SetConfigURLResp" type="tns:GetSetConfigURLRespType"/>

<element name="GetUsedConfigURL" type="tns:ReqType"/>
<element name="EventUsedConfigURL" type="tns:UsedConfigURLsType"/>
<element name="GetUsedConfigURLResp" type="tns:GetConfigURLRespType"/>

<element name="GetCoreDumpURL" type="tns:ReqType"/>
<element name="SetCoreDumpURL" type="tns:SetCoreDumpURLType"/>
<element name="EventCoreDumpURLCnf" type="tns:CoreDumpURLType"/>
<element name="GetCoreDumpURLResp" type="tns:GetSetCoreDumpURLRespType"/>
<element name="SetCoreDumpURLResp" type="tns:GetSetCoreDumpURLRespType"/>

<element name="GetSoftwareImageURL" type="tns:ReqType"/>
<element name="SetSoftwareImageURL" type="tns:SetSoftwareImageURLType"/>
<element name="EventSoftwareImageURLCnf" type="tns:SoftwareImageURLType"/>
<element name="GetSoftwareImageURLResp" type="tns:GetSetSoftwareImageURLRespType"/>
<element name="SetSoftwareImageURLResp" type="tns:GetSetSoftwareImageURLRespType"/>

<element name="GetPPPFirmwareURL" type="tns:ReqType"/>
<element name="SetPPPFirmwareURL" type="tns:SetPPPFirmwareURLType"/>
<element name="EventPPPFirmwareURLCnf" type="tns:PPPFirmwareURLType"/>
<element name="GetPPPFirmwareURLResp" type="tns:GetSetPPPFirmwareURLRespType"/>
<element name="SetPPPFirmwareURLResp" type="tns:GetSetPPPFirmwareURLRespType"/>

<element name="GetNTPServer" type="tns:ReqType"/>
<element name="SetNTPServer" type="tns:SetNTPServerType"/>
<element name="EventNTPServerCnf" type="tns:NTPServerType"/>
<element name="GetNTPServerResp" type="tns:GetSetNTPServerRespType"/>
<element name="SetNTPServerResp" type="tns:GetSetNTPServerRespType"/>

```

```

<element name="GetSysToneScheme" type="tns:ReqType"/>
<element name="SetSysToneScheme" type="tns:SetSysToneSchemeType"/>
<element name="EventSysToneSchemeCnf" type="tns:SysToneSchemeTypeType"/>
<element name="GetSysToneSchemeResp" type="tns:GetSetSysToneSchemeRespType"/>
<element name="SetSysToneSchemeResp" type="tns:GetSetSysToneSchemeRespType"/>

<element name="GetSysVoiceboxNum" type="tns:ReqType"/>
<element name="SetSysVoiceboxNum" type="tns:SetSysVoiceboxNumType"/>
<element name="EventSysVoiceboxNumCnf" type="tns:SysVoiceboxNumType"/>
<element name="GetSysVoiceboxNumResp" type="tns:GetSetSysVoiceboxNumRespType"/>
<element name="SetSysVoiceboxNumResp" type="tns:GetSetSysVoiceboxNumRespType"/>

<element name="GetTimeZone" type="tns:ReqType"/>
<element name="SetTimeZone" type="tns:SetTimeZoneType"/>
<element name="EventTimeZoneCnf" type="tns:TimeZoneType"/>
<element name="GetTimeZoneResp" type="tns:GetSetTimeZoneRespType"/>
<element name="SetTimeZoneResp" type="tns:GetSetTimeZoneRespType"/>

<element name="GetTimeZoneDetails" type="tns:ReqType"/>
<element name="SetTimeZoneDetails" type="tns:SetTimeZoneDetailsType"/>
<element name="EventTimeZoneDetailsCnf" type="tns:TimeZoneDetailsType"/>
<element name="GetTimeZoneDetailsResp" type="tns:GetSetTimeZoneDetailsRespType"/>
<element name="SetTimeZoneDetailsResp" type="tns:GetSetTimeZoneDetailsRespType"/>

<element name="GetTimeZoneList" type="tns:ReqType"/>
<element name="GetTimeZoneListResp" type="tns:GetTimeZoneListRespType"/>
<element name="EventTimeZoneList" type="tns:EventTimeZoneListType"/>

<element name="GetUserDataServer" type="tns:ReqType"/>
<element name="SetUserDataServer" type="tns:SetUserDataServerType"/>
<element name="EventUserDataServerCnf" type="tns:UserDataServerType"/>
<element name="GetUserDataServerResp" type="tns:GetSetUserDataServerRespType"/>
<element name="SetUserDataServerResp" type="tns:GetSetUserDataServerRespType"/>

<element name="GetPreserveUserDeviceSyncOMM" type="tns:ReqType"/>
<element name="SetPreserveUserDeviceSyncOMM" type="tns:SetPreserveUserDeviceSyncOMMType"/>
<element name="EventPreserveUserDeviceSyncOMMCnf" type="tns:PreserveUserDeviceSyncOMMType"/>
<element name="GetPreserveUserDeviceSyncOMMResp" type="tns:GetSetPreserveUserDeviceSyncOMMRespType"/>
<element name="SetPreserveUserDeviceSyncOMMResp" type="tns:GetSetPreserveUserDeviceSyncOMMRespType"/>

<element name="GetPreserveUserDeviceRelation" type="tns:ReqType"/>
<element name="SetPreserveUserDeviceRelation" type="tns:SetPreserveUserDeviceRelationType"/>
<element name="EventPreserveUserDeviceRelationCnf" type="tns:PreserveUserDeviceRelationType"/>
<element name="GetPreserveUserDeviceRelationResp" type="tns:GetSetPreserveUserDeviceRelationRespType"/>
<element name="SetPreserveUserDeviceRelationResp" type="tns:GetSetPreserveUserDeviceRelationRespType"/>

<element name="GetUserMonitoring" type="tns:ReqType"/>
<element name="SetUserMonitoring" type="tns:SetUserMonitoringType"/>
<element name="EventUserMonitoringCnf" type="tns:UserMonitoringType"/>
<element name="GetUserMonitoringResp" type="tns:GetSetUserMonitoringRespType"/>
<element name="SetUserMonitoringResp" type="tns:GetSetUserMonitoringRespType"/>

<element name="GetVersions" type="tns:VersionsType"/>
<element name="GetVersionsResp" type="tns:VersionsRespType"/>

<element name="EventWLANClient" type="tns:EventWLANClientType"/>
<element name="GetWLANClients" type="tns:GetWLANClientsType"/>
<element name="GetWLANClientsResp" type="tns:GetWLANClientsRespType"/>
<element name="CreateWLANProfile" type="tns>CreateSetWLANProfileType"/>
<element name="DeleteWLANProfile" type="tns>DeleteWLANProfileType"/>
<element name="GetWLANProfile" type="tns:GetRecordType"/>
<element name="SetWLANProfile" type="tns>CreateSetWLANProfileType"/>

```

```

<element name="EventWLANProfileCnf" type="tns:EventWLANProfileType"/>
<element name="CreateWLANProfileResp" type="tns:CreateSetWLANProfileRespType"/>
<element name="DeleteWLANProfileResp" type="tns:RespType"/>
<element name="GetWLANProfileResp" type="tns:GetWLANProfileRespType"/>
<element name="SetWLANProfileResp" type="tns:CreateSetWLANProfileRespType"/>

<element name="GetWLANRegDomain" type="tns:ReqType"/>
<element name="SetWLANRegDomain" type="tns:SetWLANRegDomainType"/>
<element name="EventWLANRegDomainCnf" type="tns:WLANRegDomainType"/>
<element name="GetWLANRegDomainResp" type="tns:GetSetWLANRegDomainRespType"/>
<element name="SetWLANRegDomainResp" type="tns:GetSetWLANRegDomainRespType"/>
<element name="GetWLANRegDomainList" type="tns:ReqType"/>
<element name="GetWLANRegDomainListResp" type="tns:GetWLANRegDomainListRespType"/>

<element name="CreateXMLApplication" type="tns:CreateSetXMLApplicationType"/>
<element name="DeleteXMLApplication" type="tns>DeleteXMLApplicationType"/>
<element name="GetXMLApplication" type="tns:GetRecordType"/>
<element name="SetXMLApplication" type="tns:CreateSetXMLApplicationType"/>
<element name="EventXMLApplicationCnf" type="tns:EventXMLApplicationType"/>
<element name="CreateXMLApplicationResp" type="tns:CreateSetXMLApplicationRespType"/>
<element name="DeleteXMLApplicationResp" type="tns:RespType"/>
<element name="GetXMLApplicationResp" type="tns:GetXMLApplicationRespType"/>
<element name="SetXMLApplicationResp" type="tns:CreateSetXMLApplicationRespType"/>

<element name="GenerateHealthStateAlarmTriggers" type="tns:ReqType"/>
<element name="GenerateHealthStateAlarmTriggersResp" type="tns:RespType"/>

<element name="LimitsResp" type="tns:LimitsRespType"/>
<element name="Limits" type="tns:ReqType"/>

<element name="ManualDBBackup" type="tns:ManualDBBackupRestoreType"/>
<element name="ManualDBRestore" type="tns:ManualDBBackupRestoreType"/>
<element name="ManualDBBackupResp" type="tns:ManualDBBackupRestoreRespType"/>
<element name="ManualDBRestoreResp" type="tns:ManualDBBackupRestoreRespType"/>

<element name="ManualUserDataImportResp" type="tns:RespType"/>
<element name="ManualUserDataImport" type="tns:ManualUserDataImportType"/>

<element name="DBBackupToUSB" type="tns:ReqType"/>
<element name="DBBackupToUSBResp" type="tns:RespType"/>
<element name="DBRestoreFromUSB" type="tns:ReqType"/>
<element name="DBRestoreFromUSBResp" type="tns:RespType"/>

<element name="CreateConference" type="tns:CreateConferenceType"/>
<element name="DeleteConference" type="tns>DeleteConferenceType"/>
<element name="AddUserToConference" type="tns:AddUserToConferenceType"/>
<element name="ChangeUserInConference" type="tns:ChangeUserInConferenceType"/>
<element name="DeleteUserFromConference" type="tns>DeleteUserFromConferenceType"/>
<element name="EventAddUserToConference" type="tns:EventAddChangeUserInConferenceType"/>
<element name="EventChangeUserInConference" type="tns:EventAddChangeUserInConferenceType"/>
<element name="EventCreateConference" type="tns:EventCreateConferenceType"/>
<element name="EventDeleteConference" type="tns:EventDeleteConferenceType"/>
<element name="EventDeleteUserFromConference" type="tns:EventDeleteUserFromConferenceType"/>
<element name="EventConferenceRequest" type="tns:EventConferenceRequestType"/>
<element name="EventConferenceRelease" type="tns:EventConferenceReleaseType"/>
<element name="EventRTPConferenceStreamChg" type="tns:EventRTPConferenceStreamChg"/>
<element name="SetRTPConferenceStreamChg" type="tns:SetRTPConferenceStreamChg"/>
<element name="SetRTPConferenceStreamChgResp" type="tns:RespType"/>
<element name="RequestConferenceConfirmation" type="tns:RequestConferenceConfirmationType"/>
<element name="RequestConferenceConfirmationResp" type="tns:RespType"/>
<element name="GetG729ChannelsForConference" type="tns:GetG729ChannelsForConferenceType"/>
<element name="GetG729ChannelsForConferenceResp" type="tns:GetG729ChannelsForConferenceRespType"/>

```



```

<element name="CreateConferenceResp" type="tns:CreateConferenceRespType"/>
<element name="DeleteConferenceResp" type="tns:RespType"/>
<element name="AddUserToConferenceResp" type="tns:RespType"/>
<element name="ChangeUserInConferenceResp" type="tns:RespType"/>
<element name="DeleteUserFromConferenceResp" type="tns:RespType"/>
<element name="GetReadyForConferencing" type="tns:ReqType"/>
<element name="GetReadyForConferencingResp" type="tns:GetReadyForConferencingRespType"/>
<element name="EventReadyForConferencing"/>
<element name="GetFreeConferenceChannels" type="tns:ReqType"/>
<element name="GetFreeConferenceChannelsResp" type="tns:GetFreeConferenceChannelsRespType"/>
<element name="EventFreeConferenceChannels" type="tns:FreeConferenceChannelsType"/>

<element name="CreateConferenceRoom" type="tns:CreateSetConferenceRoomType"/>
<element name="DeleteConferenceRoom" type="tns>DeleteConferenceRoomType"/>
<element name="GetConferenceRoom" type="tns:GetRecordType"/>
<element name="SetConferenceRoom" type="tns:CreateSetConferenceRoomType"/>
<element name="EventConferenceRoomCnf" type="tns:EventConferenceRoomCnfType"/>
<element name="CreateConferenceRoomResp" type="tns:CreateSetConferenceRoomRespType"/>
<element name="DeleteConferenceRoomResp" type="tns:RespType"/>
<element name="SetConferenceRoomResp" type="tns:GetConferenceRoomRespType"/>
<element name="SetConferenceRoomResp" type="tns:CreateSetConferenceRoomRespType"/>

<element name="CreateBluetoothBeacon" type="tns:CreateSetBluetoothBeaconType"/>
<element name="DeleteBluetoothBeacon" type="tns>DeleteBluetoothBeaconType"/>
<element name="GetBluetoothBeacon" type="tns:GetBluetoothBeaconType"/>
<element name="CreateSetBluetoothBeacon" type="tns:CreateSetBluetoothBeaconType"/>
<element name="EventBluetoothBeaconCnf" type="tns:EventBluetoothBeaconCnfType"/>
<element name="GetBluetoothBeaconResp" type="tns:GetBluetoothBeaconRespType"/>
<element name="SetBluetoothBeaconResp" type="tns:CreateSetBluetoothBeaconRespType"/>
<element name="CreateBluetoothBeaconResp" type="tns:CreateSetBluetoothBeaconRespType"/>
<element name="DeleteBluetoothBeaconResp" type="tns:RespType"/>
<element name="GetBluetoothBeaconSummary" type="tns:ReqType"/>
<element name="EventBluetoothBeaconSummary" type="tns:BluetoothBeaconSummaryType"/>
<element name="GetBluetoothBeaconSummaryResp" type="tns:GetBluetoothBeaconSummaryRespType"/>

<element name="GetBluetoothClientStatistic" type="tns:ReqType"/>
<element name="EventBluetoothClientStatistic" type="tns:EventBluetoothClientStatisticType"/>
<element name="GetBluetoothClientStatisticResp" type="tns:GetBluetoothClientStatisticRespType"/>

<element name="GetBluetoothSensitivity" type="tns:ReqType"/>
<element name="SetBluetoothSensitivity" type="tns:SetBluetoothSensitivityType"/>
<element name="EventBluetoothSensitivityCnf" type="tns:EventBluetoothSensitivityCnfType"/>
<element name="GetBluetoothSensitivityResp" type="tns:GetSetBluetoothSensitivityRespType"/>
<element name="SetBluetoothSensitivityResp" type="tns:GetSetBluetoothSensitivityRespType"/>

<element name="GetBluetoothGlobalSettings" type="tns:ReqType"/>
<element name="SetBluetoothGlobalSettings" type="tns:SetBluetoothGlobalSettingsType"/>
<element name="EventBluetoothGlobalSettingsCnf" type="tns:EventBluetoothGlobalSettingsCnfType"/>
<element name="GetBluetoothGlobalSettingsResp" type="tns:GetSetBluetoothGlobalSettingsRespType"/>
<element name="SetBluetoothGlobalSettingsResp" type="tns:GetSetBluetoothGlobalSettingsRespType"/>

<element name="CreateVideoDev" type="tns:CreateSetVideoDevType"/>
<element name="DeleteVideoDev" type="tns>DeleteGetVideoDevType"/>
<element name="GetVideoDev" type="tns>DeleteGetVideoDevType"/>
<element name="SetVideoDev" type="tns:CreateSetVideoDevType"/>
<element name="EventVideoDevCnf" type="tns:VideoDevType"/>
<element name="EventVideoDevLink" type="tns:VideoDevLinkType"/>
<element name="GetVideoDevResp" type="tns:CreateGetSetVideoDevRespType"/>
<element name="SetVideoDevResp" type="tns:CreateGetSetVideoDevRespType"/>
<element name="CreateVideoDevResp" type="tns:CreateVideoDevRespType"/>
<element name="GetVideoDevResp" type="tns:CreateGetSetVideoDevRespType"/>
<element name="DeleteVideoDevResp" type="tns:RespType"/>

```

```

<element name="GetVideoDevLink" type="tns:GetVideoDevLinkType"/>
<element name="GetVideoDevLinkResp" type="tns:RespType"/>
<element name="GetVideoDevSummary" type="tns:ReqType"/>
<element name="EventVideoDevSummary" type="tns:VideoDevSummaryType"/>
<element name="GetVideoDevSummaryResp" type="tns:GetVideoDevSummaryRespType"/>

<element name="GetAdditionalSettings" type="tns:ReqType"/>
<element name="SetAdditionalSettings" type="tns:SetAdditionalSettingsType"/>
<element name="EventAdditionalSettingsCnf" type="tns:AdditionalSettings"/>
<element name="GetAdditionalSettingsResp" type="tns:GetSetAdditionalSettingsRespType"/>
<element name="SetAdditionalSettingsResp" type="tns:GetSetAdditionalSettingsRespType"/>

<element name="GetPpProfile" type="tns:GetPpProfileType"/>
<element name="SetPpProfile" type="tns:SetPpProfileType"/>
<element name="EventPpProfileCnf" type="tns:EventPpProfileType"/>

<element name="GetPpProfileResp" type="tns:GetSetPpProfileRespType"/>
<element name="SetPpProfileResp" type="tns:GetSetPpProfileRespType"/>

<element name="OpenResp" type="tns:OpenRespType"/>
<element name="Open" type="tns:OpenType"/>

<element name="PingResp" type="tns:PingRespType"/>
<element name="Ping" type="tns:PingReqType"/>

<element name="PutFileResp" type="tns:RespType"/>
<element name="PutFile" type="tns:PutFileType"/>

<element name="RequestPositionInfoResp" type="tns:RespType"/>
<element name="RequestPositionInfo" type="tns:RequestPositionInfoType"/>

<element name="ResetRFPStatisticResp" type="tns:RespType"/>
<element name="ResetRFPStatistic" type="tns:ResetRFPStatisticType"/>

<element name="ResetSysStatisticResp" type="tns:RespType"/>
<element name="ResetSysStatistic" type="tns:ResetSysStatisticType"/>

<element name="ResetTimeZoneDetailsResp" type="tns:ReqType"/>
<element name="ResetTimeZoneDetails" type="tns:RespType"/>

<element name="Subscribe" type="tns:SubscribeType"/>
<element name="SubscribeResp" type="tns:SubscribeRespType"/>

<element name="SystemRestartResp" type="tns:RespType"/>
<element name="SystemRestart" type="tns:SystemRestartType"/>

</schema>

```



© Copyright 2015, Mitel Networks Corporation. All Rights Reserved. The Mitel word and logo are trademarks of Mitel Networks Corporation. Any reference to third party trademarks are for reference only and Mitel makes no representation of ownership of these marks.