(C) UTC Fire & Security 2013

ATS1000A message reference

Protocol revision 23.23.0

Document revision 4057

Table of contents

Introduction....15
Context....20
fault....22
return.void....24
return.short....25
return.bool....26
device.panelId....28
return.panelId....29
insert.panelId....30
device.getDescription....31
device.Description....32
begin.changeSessionKey....38
end.changeSessionKey....39
return.changeSessionKey....40
device.getConnect....41
device.disconnect....43
is.Alive....44
open.LOG....45
close.LOG....46
select.getLOG....47
return.getLOG....48
start.MONITOR....61
stop.MONITOR....62
pause.MONITOR....63
msg.MONITOR....64
msgCOS.ALL....77
msgCOS.CAM_RANGETST....82
msgCOS.SYS_INV_WALKTST_REP....86
get.timedate....87
return.timedate....88
get.privileges....89
return.privileges....90
get.UserInfo....95
return.UserInfo....96
get.liveEvents....99
return.sysevent....103
getCOS.ZONE....109
returnCOS.ZONE....110
getCOS.AREA....112
returnCOS.AREA....113
getCOS.RAS....114
returnCOS.RAS....115

getCOS.DGP..... 116
returnCOS.DGP.....117
getCOS.OUT..... 118
returnCOS.OUT.....119
getCOS.FILTER.....120
returnCOS.FILTER..... 121
getCOS.PCC..... 123
returnCOS.PCC.....124
getCOS.CS.....126
returnCOS.CS.....127
getCOS.TRIGG.....129
returnCOS.TRIGG.....130
getCOS.USER..... 131
returnCOS.USER.....132
getCOS.UG.....133
returnCOS.UG..... 134
getCOS.EXCP..... 135
returnCOS.EXCP.....136
getCOS.FOB..... 138
returnCOS.FOB.....139
getCOS.CAMERA.....140
returnCOS.CAMERA..... 141
getSTAT.ZONE.....142
returnSTAT.ZONE.....143
getSTAT.AREA.....147
returnSTAT.AREA.....148
getSTAT.RAS..... 161
returnSTAT.RAS.....162
getSTAT.DGP..... 165
returnSTAT.DGP.....166
getSTAT.DGP0.....169
returnSTAT.DGP0.....170
getSTAT.OUT..... 174
returnSTAT.OUT.....175
getSTAT.FILTER.....176
returnSTAT.FILTER..... 177
getSTAT.PCC..... 178
returnSTAT.PCC.....179
getSTAT.SYS..... 180
returnSTAT.SYS.....181
getSTAT.CS..... 184
returnSTAT.CS.....185
getSTAT.TRIGG.....186
returnSTAT.TRIGG..... 187
getSTAT.USER.....189
returnSTAT.USER.....190

getSTAT.UG..... 192
returnSTAT.UG..... 193
getSTAT.EXCP..... 194
returnSTAT.EXCP..... 195
getSTAT.SCAL..... 196
returnSTAT.SCAL..... 197
getSTAT.FOB..... 199
returnSTAT.FOB..... 200
getSTAT.CAMERA..... 202
returnSTAT.CAMERA..... 203
createCC.A_STATE..... 204
createCC.A_SET..... 207
createCC.A_CONFAL..... 212
createCC.A_WALKTST..... 215
createCC.A_UNSET..... 218
createCC.A_PARTSET..... 222
createCC.A_PARTSET2..... 227
createCC.ZONE..... 232
createCC.DEVICE..... 235
createCC.OUT_TRIG..... 238
createCC.OUTPUT..... 241
createCC.ENG_RES..... 244
createCC.TIME_DATE..... 247
createCC.PC_CONN..... 250
createCC.USER..... 253
createCC.TEST_CALL..... 256
createCC.PICTURES..... 259
createCC.CAM_RANGETST..... 261
createCC.SYS_INV_WALKTST..... 264
createCC.SYS_WALKTST_MODE..... 267
destroyCC.SESSION..... 270
statusCC.SESSION..... 271
return.statusCC..... 272
fnCC.CAM_RANGETST_START..... 275
fnCC.CAM_RANGETST_ADDCAM..... 276
fnCC.A_STATE_GET_INH..... 277
fnCC.A_STATE_GET_UNINH..... 278
fnCC.A_SET_SETAREAS..... 279
fnCC.A_SET_GETFAULT..... 280
fnCC.A_SET_GETACTIVE..... 281
fnCC.A_SET_GETINHIB..... 282
fnCC.A_SET_INHFAULT..... 283
fnCC.A_SET_INHACTIVE..... 284
fnCC.A_SET_FORCEDSET..... 285
fnCC.A_CONFAL_START..... 286
fnCC.A_CONFAL_GETALARM..... 287

fnCC.A_CONFAL_CONFALARM..... 288
fnCC.A_WALKTST_START..... 289
fnCC.A_WALKTST_GETLIST..... 290
fnCC.A_WALKTST_GETEV..... 291
fnCC.A_WALKTST_GETRES..... 292
fnCC.A_WALKTST_GET_WARN_TIME..... 293
fnCC.SYS_INV_WALKTST_REP..... 294
fnCC.SYS_INV_WALKTST_RESET..... 295
fnCC.SYS_CHANGE_WALKTST_MODE..... 296
fnCC.A_WALKTST_START_WITH_REP..... 297
fnCC.A_WALKTST_ADD_ZONE..... 298
fnCC.A_UNSET_UNSETAREAS..... 299
fnCC.A_UNSET_SKIP..... 300
fnCC.A_UNSET_GETALARM..... 301
fnCC.A_UNSET_CONFALARM..... 302
fnCC.A_UNSET_GETFAULT..... 303
fnCC.A_UNSET_CONFFAULT..... 304
fnCC.ZONE_ISOLATE..... 305
fnCC.ZONE_UNISOLATE..... 306
fnCC.ZONE_INHIBIT..... 307
fnCC.ZONE_UNINHIBIT..... 308
fnCC.DEVICE_ISOLATE..... 309
fnCC.DEVICE_UNISOLATE..... 311
fnCC.BATTERY_TEST_START..... 313
fnCC.BATTERY_TEST_CANCEL..... 315
fnCC.OUT_TRIG_ACTIVATE..... 317
fnCC.OUT_TRIG_DEACTIVATE..... 318
fnCC.OUTPUT_ACTIVATE..... 319
fnCC.OUTPUT_DEACTIVATE..... 320
fnCC.ENG_RES_DORESET..... 321
fnCC.ENG_RES_GETRESULT..... 322
fnCC.ENG_RES_GETCODE..... 323
fnCC.TIME_DATE_SET..... 324
fnCC.PC_CONN_START..... 325
fnCC.PC_CONN_STOP..... 326
fnCC.USER_SETCONTROL..... 327
fnCC.USER_SETREPORT..... 328
fnCC.USER_GETPHONE..... 329
return.UserPhone..... 330
fnCC.USER_SETPHONE..... 331
fnCC.USER_SETPIN..... 332
fnCC.OUT_SCHED_TRIG_ACTIVATE..... 333
fnCC.OUT_SCHED_TRIG_DEACTIVATE..... 334
fnCC.TEST_CALL_START..... 335
fnCC.TEST_CALL_STATUS..... 336
select.ZoneNames..... 337

return.ZoneNames..... 338
select.AreaNames..... 339
return.AreaNames..... 340
select.RASNames.....341
return.RASNames.....342
select.DGPNames.....343
return.DGPNames.....344
select.UserNames..... 345
return.UserNames..... 346
select.OutputNames.....347
return.OutputNames.....348
select.CEvFilterNames.....349
return.CEvFilterNames.....350
select.UserGroupNames.....351
return.UserGroupNames.....352
select.CSNames.....353
return.CSNames.....354
select.DLNames.....355
return.DLNames.....356
select.SYSNames.....357
return.SYSNames.....358
select.PCCNames.....359
return.PCCNames.....360
select.TriggerNames.....361
return.TriggerNames.....362
select.SchedActNames.....363
return.SchedActNames.....364
select.SchedActLstNames..... 365
return.SchedActLstNames..... 366
select.SchedExcNames.....367
return.SchedExcNames.....368
select.ScheduleNames.....369
return.ScheduleNames.....370
select.FobNames.....371
return.FobNames.....372
select.CameraNames.....373
return.CameraNames.....374
select.Zone..... 375
insert.Zone..... 376
return.Zone..... 414
return.Area..... 423
insert.Area..... 427
select.Area..... 431
select.User..... 432
return.User..... 433
insert.User..... 437

add.Users....441
add.RemoteUsers....445
select.CS....446
selectV.CS....447
return.CS....448
return.CS_2....453
return.CS_CMN....459
return.CS_PHONE....465
return.CS_IP....466
return.CS_USER....467
return.CS_USERGROUP....468
insert.CS....469
insert.CS_2....474
insert.CS_CMN....480
insertV.CS_PHONE....487
insertV.CS_IP....488
insertV.CS_USER....489
insertV.CS_USERGROUP....490
select.RAS....491
selectV.RASAct....492
return.RAS....493
insert.RAS....503
insertV.RASActNone....513
insertV.RASActSet....514
insertV.RASActUnset....517
insertV.RASActTrigger....519
insertV.RASActDoorbell....520
insertV.RASActPSet1....523
insertV.RASActPSet2....526
insertV.RASActInh....529
insertV.RASActTCall....530
insertV.RASActPCC....531
insertV.RASActServIn....532
insertV.RASActPanic....533
insertV.RASActDoorbellRAS....534
insertV.RASActSetWET....537
insertV.RASActFireReset....540
insertV.RASActOpenZn....542
insertV.RASActAlarmZn....543
insertV.RASActFaults....544
insertV.RASActAlarmMem....545
insertV.RASActZonesAck....546
insertV.RASActWalkTest....547
insertV.RASActOutputTest....549
insertV.RASActFire....551
insertV.RASActMedical....552

insertV.RASActTakePicture....553
select.FOB....554
selectV.FobAct....555
return.FOB....556
insert.FOB....558
insertV.FOBActNone....560
insertV.FOBActSet....561
insertV.FOBActUnset....563
insertV.FOBActTrigger....565
insertV.FOBActPSet1....566
insertV.FOBActPSet2....568
insertV.FOBActPanic....570
insertV.FOBActTakePicture....571
select.Camera....572
return.Camera....573
delete.Camera....580
insert.Camera....581
select.DGP....588
return.DGP....589
insert.DGP....592
select.Output....595
return.Output....596
insert.Output....599
select.Trigger....602
return.Trigger....603
insert.Trigger....604
select.UserGroup....605
return.UserGroup....606
insert.UserGroup....617
add.UserGroups....628
select.DL....639
selectV.DL....640
selectV.DL_MMS....641
select.DL_INFO....642
return.DL_INFO....643
return.DL....644
return.DL_PSTN....647
return.DL_ISDN....650
return.DL_GSM....653
return.DL_IP....658
return.DL_STEL....663
return.DL_CHIRON....665
return.DL_75XX....667
return.DL_VEMPTY....669
return.DL_GSM_IP....670
return.DL_GSM_MMS....673

insert.DL....675
select.SiaEvent....678
insert.SiaEvent....681
return.SiaEvent....688
select.PCC....695
selectV.PCC....696
return.PCC....697
return.PCC_2....699
return.PCC_CMN....702
return.PCC_PHONE....704
return.PCC_IP....705
insert.PCC....706
insert.PCC_2....708
insert.PCC_CMN....711
insertV.PCC_PHONE....713
insertV.PCC_IP....714
return.CommandStatus....715
select.CEvFilter....716
return.CEvFilter....717
insert.CEvFilter....766
delete.Zone....815
delete.RAS....816
delete.FOB....817
delete.DGP....818
delete.User....819
delete.Output....820
delete.UserGroup....821
delete.CEvFilter....822
deleteM.Zone....823
deleteM.RAS....826
deleteM.FOB....829
deleteM.Camera....832
deleteM.DGP....835
deleteM.User....838
deleteM.Output....841
deleteM.UserGroup....844
deleteM.CEvFilter....847
getValid.Areas....850
getValid.Cameras....851
return.validAreas....852
return.validCameras....853
getAvailM.Zone....854
getAvailM.RAS....855
getAvailM.DGP....856
getAvailM.User....857
getAvailM.Output....858

getAvailM.CEvFilter....859
getAvailM.UserGroup....860
getAvailM.FOB....861
getAvailM.Camera....862
blockID.Zone....863
blockID.Area....864
blockID.RAS....865
blockID.DGP....866
blockID.User....867
blockID.Output....868
blockID.Trigger....869
blockID.CEvFilter....870
blockID.UserGroup....871
blockID.CS....872
blockID.DL....873
blockID.SYS....874
blockID.PCC....875
blockID.SiaEvent....876
blockID.SchedAct....877
blockID.SchedActLst....878
blockID.SchedExc....879
blockID.Schedule....880
blockID.Fob....881
blockID.Camera....882
blockID.RemoteUser....883
blockID.Master....884
blockIDM.Zone....885
blockIDM.Area....886
blockIDM.RAS....887
blockIDM.DGP....888
blockIDM.User....889
blockIDM.Output....890
blockIDM.Trigger....891
blockIDM.CEvFilter....892
blockIDM.UserGroup....893
blockIDM.CS....894
blockIDM.DL....895
blockIDM.SYS....896
blockIDM.PCC....897
blockIDM.SiaEvent....898
blockIDM.SchedAct....899
blockIDM.SchedActLst....900
blockIDM.SchedExc....901
blockIDM.Schedule....902
blockIDM.Fob....903
blockIDM.Camera....904

blockIDM.RemoteUser....905
return.AvailMZone....906
return.AvailMRAS....907
return.AvailMDGP....908
return.AvailMUser....909
return.AvailMOutput....910
return.AvailMCevFilter....911
return.AvailMUserGroup....912
return.AvailMFob....913
return.AvailMCamera....914
return.BlockId....915
return.BlockIdMaster....916
return.BlockIdZoneM....919
return.BlockIdAreaM....920
return.BlockIdRASM....921
return.BlockIdDGPM....922
return.BlockIdUserM....923
return.BlockIdOutputM....924
return.BlockIdTriggerM....925
return.BlockIdCevFilterM....926
return.BlockIdUserGroupM....927
return.BlockIdCSM....928
return.BlockIdDLM....929
return.BlockIdSysM....930
return.BlockIdPCCM....931
return.BlockIdSiaEventM....932
return.BlockIdSchedActM....933
return.BlockIdSchedActLstM....934
return.BlockIdSchedExcM....935
return.BlockIdScheduleM....936
return.BlockIdFobM....937
return.BlockIdCameraM....938
return.BlockIdRemoteUserM....939
select.SYS1....940
return.SYS1....941
insert.SYS1....947
select.SYS2....953
return.SYS2....954
insert.SYS2....962
select.SYS3....970
return.SYS3....971
insert.SYS3....989
select.SYS0....1007
return.SYS0....1008
select.SYS4....1011
return.SYS4....1012

insert.SYS4 1015
device.SecondPIN 1018
insert.putPIN 1019
insert.putRemotePIN 1020
insert.putCARD 1021
select.getPIN 1022
select.getRemotePIN 1023
select.getCARD 1024
return.getPIN 1025
return.getCARD 1026
return.getRemotePIN 1027
generate.userPIN 1028
generate.userRemotePIN 1029
start.Users 1030
stop.Users 1031
commit.Users 1032
select.CSAccount 1033
return.CSAccount 1034
return.CSAccount2 1035
insert.CSAccount 1037
insert.CSAccount2 1038
return.DL_2 1040
insert.DL_2 1045
insert.DL_PSTN 1050
insert.DL_ISDN 1053
insert.DL_GSM 1056
insert.DL_IP 1061
insert.DL_STEL 1066
insert.DL_CHIRON 1068
insert.DL_75XX 1070
insertV.DL_GSM_IP 1072
insertV.DL_GSM_MMS 1075
begin.InitKey 1077
insert.InitKey 1078
select.SchedAct 1080
selectV.SACTAct 1081
select.SchedActLst 1082
select.SchedExc 1083
select.Schedule 1084
select.ScheduleDayActions 1085
select.Schedule2 1086
insert.SchedAct 1087
insertV.SACTActSet 1089
insertV.SACTActUnset 1091
insertV.SACTActTrigger 1093
insertV.SACTActDoorbell 1094

insertV.SACTActUGMask....1097
insertV.SACTActRASControl....1104
insertV.SACTActPSet1....1105
insertV.SACTActPSet2....1107
insert.SchedActLst....1109
insert.SchedExc....1112
insert.Schedule....1118
insert.ScheduleDayActions....1137
insert.Schedule2....1141
return.SchedAct....1156
return.ActNone....1158
return.ActSet....1159
return.ActUnset....1161
return.ActTrigger....1163
return.ActDoorbell....1164
return.ActUGMask....1166
return.ActTakePicture....1173
return.ActRASControl....1174
return.ActPSet1....1175
return.ActPSet2....1177
return.ActInh....1179
return.ActTCall....1180
return.ActPCC....1181
return.ActServIn....1182
return.ActPanic....1183
return.ActDoorbellRAS....1184
return.ActSetWET....1186
return.ActFireReset....1188
return.ActOpenZn....1190
return.ActAlarmZn....1191
return.ActFaults....1192
return.ActAlarmMem....1193
return.ActZonesAck....1194
return.ActWalkTest....1195
return.ActOutputTest....1197
return.ActFire....1199
return.ActMedical....1200
return.SchedActLst....1201
return.SchedExc....1204
return.Schedule....1210
return.ScheduleDayActions....1229
return.Schedule2....1233
prepareEnc.UPLOAD....1248
startEnc.UPLOAD....1249
event.Aggregate....1250
prepareEnc.DOWNLOAD....1251

call.Aggregate....1252
return.Aggregate.... 1253
finishedEnc.UPLOAD.... 1254
cancelEncUPLOAD....1256
event.PebMemoryStatus....1257
event.PebMemoryClearProgress....1258
return.GetPebs....1259
return.GetPebCameras....1261
msgCamera.GetPEBs.... 1264
msgCamera.GetPebCameras.... 1265
event.CameraDir....1266
event.PictureChunk.... 1268
sesCamera.prepareCameraDir....1269
sesCamera.prepareLoadPicture....1270
sesCamera.prepareTakePicture....1271
sesCamera.prepareMemoryClear....1272
sesCamera.prepareMemoryInfo....1273
sesCamera.start....1274
sesCamera.completed....1275
sesCamera.cancel.... 1277

Introduction

This document lists all messages supported by the ATS Communication Engine for communication to Ats Advanced control panels.

Every message definition begins with the message identifier string, for example `device.getDescription`. The direction tells you whether the message can be sent from the PC to the device (*output*), from the device to the PC (*input*) or both ways (*both*). After that, message description and notes follow, which explain the purpose of the message and provide additional information.

Then the list of the message's properties follows. Each property is described by its name, type (possibly with value constraints) and multiplicity indication. Please pay special attention to the constraint *nullable*. If present in the property definition it means that null can be assigned to the property, in which case the property will be considered as not set to any value. The *multiplicity* indicates whether the property is a scalar or a vector.

Remote access level

The Ats Advanced panel allows to connect remotely for various access level:

UPLOAD
: Read configuration data from the panel.
This access is restricted only for *Installer*.

DOWNLOAD
: Write configuration data from the panel.
This access is restricted only for *Installer*. It requires exclusive access to the system. It means that if this access level is used other remote/local requests are blocked (*Installer* on the system). It cannot be used with *CONTROL* or *MONITOR*.

LOG READ
: Read log contents.
Access to events are limited by available areas. Access to functions are limited by user privileges.

MONITOR
: Monitor the system, online log read.
This access is restricted only for *Installer* or *Supervisor*. Access to events are limited by available areas. It can be used with all other access levels except *DOWNLOAD*. This access is available since protocol version *011*.

DIAGNOSE

Examine system status.
Access to events are limited by available areas. It can be used with all other access levels except *DOWNLOAD*. This access is available since protocol version *011*.

CONTROL

Control the system.
Access to functions are limited by available areas and user privileges. This access is available since protocol version *011*.

Remote session

The typical remote session consists of the following stages:

1. **DISCOVER** — Detecting the connected panel.
   This involves the `device.getDescription` request and analizing of the received `device.Description` response.
2. **SET SESSION KEY** — Establishing encrypted connection.
   This two stage process allows to negotiate session enryption key being used during the remote session. This is performed with `begin.changeSessionKey` and `end.changeSessionKey` requests. This stage is available and mandatory for encrypted communication channel.
3. **LOGIN** — Login user.
   The `device.getConnect` request allows to specify *PIN* and remote access level of the user.
4. **ACTIVE** — Active connection.
   In this stage all application defined tasks are performed accordingly with the required remote access level.
5. **LOGOUT** — Logout user.
   The `device.disconnect` request performs the action.

LOG READ

The *LOG READ* access level is used to read events from the system log.

The following requests are available to perform the reading logged events:

- `open.LOG`
- `select.getLOG`
- `close.LOG`

MONITOR

The *MONITOR* access level allows to read asynchronously events from the remote station.

Log events are transmitted to the remote station in the same format as stored in the system events log. System keeps track of sent events and filter the output so only events related to available areas are transmitted.

The following requests control the log events sending process:

- `start.MONITOR`
- `stop.MONITOR`
- `pause.MONITOR`

The events are received with `msg.MONITOR` message.

DIAGNOSE

The *DIAGNOSE* access level is used to determine system status.

The `msgCOS.ALL` event is received from the panel asynchronously whenever something is changed in the systems state.

The `msgCOS.CAM_RANGETST` event is received from the panel asynchronously whenever new camera range test data is available (during appropriate CC session).

The `msgCOS.SYS_INV_WALKTST_REP` event is received from the panel asynchronously while report inverted walk test report is being generated (during appropriate CC session).

Additionally the following informational requests are supported in *DIAGNOSE* access level:

- `get.liveEvents`
- `get.timedate`
- `get.privileges`
- `get.UserInfo`

CONTROL

The *CONTROL* access level is used to remote control.

Functions

Control functions are serviced as separate tasks that are controlled by a set of special requests. Basic set of methods to control the task should include:

- Create request to initialize and start the task.
- Destroy request to stop or abort the task.
- Status request to get the current state of the task.

In addition to these basic requests set, each task provides a set of special methods to control specific functions. Note that only one active control task is allowed per user session.

Generic faults

The following set of faults (exceptions) can be generated by the control mechanism.

Symbol	Value	Description
NO_ACCESS	2	No access is possible to the required functionality or object
NO_OBJECT	3	Requested object (session) does not exist
CC_WRONG_STAT	24	Function executed in wrong C&C machine state
CC_WRONG_TYPE	25	Incorrect C&C machine type used
CC_WRONG_SESSION	26	C&C Session opened in wrong mode
CC_WRONG_AREAS	27	Incorrect list of areas used for C&C function
CC_BUSY_AREAS	28	Some of areas from list area busy (locked or in incorrect state)
CC_WRONG_PRIVILEGES	29	User does not have enough privileges to use C&C function

Generic requests

A family of *create* requests:

- `createCC.A_SET`
- `createCC.A_PARTSET`
- `createCC.A_PARTSET2`
- `createCC.A_CONFAL`
- `createCC.A_WALKTST`

- createCC.A_UNSET
- createCC.A_STATE
- createCC.ZONE
- createCC.DEVICE
- createCC.OUT_TRIG
- createCC.OUTPUT
- createCC.ENG_RES
- createCC.TIME_DATE
- createCC.PC_CONN
- createCC.USER
- createCC.TEST_CALL
- createCC.CAM_RANGETST
- createCC.SYS_INV_WALKTST

Requests common for all tasks:

- statusCC.SESSION
- destroyCC.SESSION

Context

This is a set of properties available in the message factory.

The properties are used in the messages created by the message factory for validating various constraints that depend on them.

The proper configuration of the properties is usually performed during discovering the connected panel with `device.getDescription` request while analysing the response `device.Description` and copying the apropriate values to the context properties accordingly.

`model`

multiplicity: *single (static)*
type: *integer*

Model of the panel to validate messages and its properties.

value	symbol
1	ATS1000A
2	ATS2000A
3	ATS1000AIP
4	ATS2000AIP
5	ATS1000AC
6	ATS2000AC
7	ATS1000AIPC
8	ATS2000AIPC

`protocol`

multiplicity: *single (static)*
type: *integer*

Protocol version to validate messages and its properties.

value	symbol
1	001
2	002
3	003
4	004
5	005
6	006
7	007
8	008
9	009

value	symbol
10	010
11	011
12	012
13	013
14	014
15	015
16	016
17	017
18	018
19	019
20	020
21	021
22	022
23	023

panelNorm

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Current panel norm.

value	symbol
1	EN50131
2	INCERT

fault

direction: *input*

The `fault` message is used internally to notify exceptional result (such as error) for the requests from management software.

The message is not received directly, rather it is converted to the `AtsFaultException` in the Ats Communication Engine. The value of the `faultCode` propery from the internally received message is copied to the generated exception.

faultCode

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Returns a fault code of the operation.

value	symbol
1	FAULT_METHOD_ERROR
2	FAULT_NO_ACCESS
3	FAULT_NO_OBJECT
4	FAULT_LOG_NEWEST
5	FAULT_LOG_OLDEST
6	FAULT_PIN
7	FAULT_CARD
8	FAULT_USER
9	FAULT_UG
10	FAULT_UG_DATA
11	FAULT_UG_AREAS
12	FAULT_UG_PRIVILEGES
13	FAULT_COMMIT
14	FAULT_USER_UG_NOT_EXIST
15	FAULT_INSTALLER_UG
16	FAULT_SUPERVISOR_UG
17	FAULT_USER_UG
18	FAULT_ZONE_DATA
19	FAULT_UG_EVENT_FILTER
20	FAULT_DATA_NOT_VALID
21	FAULT_OFFLINE
22	FAULT_SUPERVISOR_REQUIRED
23	FAULT_PANEL_BUSY
24	FAULT_CC_WRONG_STAT

value	symbol
25	FAULT_CC_WRONG_TYPE
26	FAULT_CC_WRONG_SESSION
27	FAULT_CC_WRONG_AREAS
28	FAULT_CC_BUSY_AREAS
29	FAULT_CC_WRONG_PRIVILEGES
30	FAULT_CC_OPERATION_CANCELED
31	FAULT_CC_COMMUNICATION_TOUT
32	FAULT_CC_BUSY_PEB
33	FAULT_CC_NRDY_PIC
34	FAULT_CC_WRONG_TSTAMP
35	FAULT_CC_OFFL_PEB
36	FAULT_CC_COMM_PEB
37	FAULT_CC_BUSY_CAMERA
38	FAULT_CC_ISOL_CAMERA
39	FAULT_CC_LIMIT_PIC
40	FAULT_CC_NOTALLOWED_PIC
41	FAULT_CC_FAULT
42	FAULT_CC_MEMORYFULL
50	FAULT_FEATURE_NOT_SUPPORTED
51	FAULT_RETRY
65535	METHOD_NOT_FOUND

return.void

direction: *input*

This is the default return message for method which does not return any value.

See also

- begin.changeSessionKey
- end.changeSessionKey
- is.Alive
- destroyCC.SESSION
- fnCC.A_STATE_GET_INH
- fnCC.A_STATE_GET_UNINH
- fnCC.A_SET_GETFAULT
- fnCC.A_SET_GETACTIVE
- fnCC.A_SET_GETINHIB
- fnCC.A_CONFAL_GETALARM
- fnCC.A_WALKTST_GETLIST
- fnCC.A_WALKTST_GETEV
- fnCC.A_WALKTST_GETRES
- fnCC.A_UNSET_SKIP
- fnCC.A_UNSET_GETALARM
- fnCC.A_UNSET_GETFAULT
- fnCC.ENG_RES_DORESET

return.short

direction: *input*

This is the default return message for method which returns short value.

See also

- createCC.A_STATE
- createCC.A_SET
- createCC.A_CONFAL
- createCC.A_WALKTST
- createCC.A_UNSET
- createCC.A_PARTSET
- createCC.A_PARTSET2
- createCC.ZONE
- createCC.DEVICE
- createCC.OUT_TRIG
- createCC.OUTPUT
- createCC.ENG_RES
- createCC.TIME_DATE
- createCC.PC_CONN
- createCC.USER
- createCC.TEST_CALL
- createCC.PICTURES
- createCC.CAM_RANGETST
- createCC.SYS_INV_WALKTST
- createCC.SYS_WALKTST_MODE
- fnCC.A_WALKTST_GET_WARN_TIME
- fnCC.ENG_RES_GETCODE
- fnCC.TEST_CALL_STATUS

result

multiplicity: *single (static)*
type: *integer*

Result value (16 bit).

return.bool

direction: *input*

This is the default return message for method which returns bool value.

See also

- start.MONITOR
- stop.MONITOR
- pause.MONITOR
- fnCC.CAM_RANGETST_START
- fnCC.CAM_RANGETST_ADDCAM
- fnCC.A_SET_SETAREAS
- fnCC.A_SET_INHFAULT
- fnCC.A_SET_INHACTIVE
- fnCC.A_SET_FORCEDSET
- fnCC.A_CONFAL_START
- fnCC.A_CONFAL_CONFALARM
- fnCC.A_WALKTST_START
- fnCC.SYS_INV_WALKTST_REP
- fnCC.SYS_INV_WALKTST_RESET
- fnCC.SYS_CHANGE_WALKTST_MODE
- fnCC.A_WALKTST_START_WITH_REP
- fnCC.A_WALKTST_ADD_ZONE
- fnCC.A_UNSET_UNSETAREAS
- fnCC.A_UNSET_CONFALARM
- fnCC.A_UNSET_CONFFAULT
- fnCC.ZONE_ISOLATE
- fnCC.ZONE_UNISOLATE
- fnCC.ZONE_INHIBIT
- fnCC.ZONE_UNINHIBIT
- fnCC.DEVICE_ISOLATE
- fnCC.DEVICE_UNISOLATE
- fnCC.BATTERY_TEST_START
- fnCC.BATTERY_TEST_CANCEL
- fnCC.OUT_TRIG_ACTIVATE
- fnCC.OUT_TRIG_DEACTIVATE
- fnCC.OUTPUT_ACTIVATE
- fnCC.OUTPUT_DEACTIVATE
- fnCC.ENG_RES_GETRESULT
- fnCC.TIME_DATE_SET
- fnCC.PC_CONN_START
- fnCC.PC_CONN_STOP
- fnCC.USER_SETCONTROL

- `fnCC.USER_SETREPORT`
- `fnCC.USER_SETPHONE`
- `fnCC.USER_SETPIN`
- `fnCC.TEST_CALL_START`

`result`

multiplicity: *single (static)*
type: *integer*

Result value (1 bit).

value	**symbol**
0	false
1	true

device.panelId

direction: *output*

This is the message to fetch custom panel identifier used typically for indentification of the device in downloader(s) database.

The received result is `return.panelId`.

Remarks:

- The message is available since protocol version *023*.

See also

- `return.panelId`

return.panelId

direction: *input*

Response with connected device identification in a request of `device.panelId` message.

Remarks:

- The message is available since protocol version *023*.

See also

- `device.panelId`

id

multiplicity: *single (static)*
type: *string*

Device account code.

insert.panelId

direction: *output*

This message is used to change the panel identifier.

The panel response for the message is `return.BlockId`.

Remarks:
- The message is available since protocol version *023*.

id

multiplicity: *single (static)*
type: *string*

Device account code.

- This string must be 0-16 characters length.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

device.getDescription

direction: *output*

This is the request to detect the connected device.

The received result is `device.Description`.

See also

- `device.Description`

device.Description

direction: *input*

Response for `device.getDescription` request.

The response can be used to properly configure context for the connection by setting the properties available in message factory.

See also

- `device.getDescription`

device.name

multiplicity: *single (static)*
type: *string*

Device name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

device.FWID_ProductName

multiplicity: *single (static)*
type: *string*

Product name.

This is the device model string that can be directly copied to the `model` property in the message factory.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

`device.FWID_FirmwareVersion`

multiplicity: *single (static)*
type: *string*

Firmware version.

The fimware version string has the format `MR_999.999.9999`.
The `MR` is purpose identfier and stands for *Market Release*. It may be replaced by `PR`, which stands for *Patch Release* and is a service pack for specific release. Then three numbers follow. The first number is a configuration database format number. The second number (substring between dots) is the protocol number. It should be used as value for `protocol` property in message factory. The last number is a build version number, and it is incremented for every service pack release or grows for every next market release with incremented protocol version.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

device.FWID_SerialNumber

multiplicity: *single (static)*
type: *string*

Serial number.

The serial number is a unique number assigned to the device during manufacturing.

The number is a formatted as 48-bit long value in hexadecimal notation.

The value participates in encrypted connections to the panel.

It is also usefull to identify the panel. I.e. when the panel is connected via *USB* to the PC then Windows operation system uses the number to recognize the device and try to assing the same virtual serial port name to it.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

device.FWID_MAC

multiplicity: *single (static)*
type: *string*

MAC address.

Remarks
- The property is enabled since protocol version *009*.
- The only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

device.FWID_EncMode

multiplicity: *single (static)*
type: *integer*

Encryption mode for the current channel.

Remarks
- The property is enabled since protocol version *009*.

value	symbol
0	NONE
1	AES-128

panelNorm

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Panel norm

Remarks
- The property is enabled since protocol version *005*.

value	symbol
1	EN50131
2	INCERT

SYSM_DST1_MTH

multiplicity: *single (static)*
type: *integer*

Daylight saving time beginning month

Remarks
- The property is enabled since protocol version *003*.

SYSM_DST2_MTH

multiplicity: *single (static)*
type: *integer*

Daylight saving time ending month

Remarks
- The property is enabled since protocol version *003*.

SYSM_DST1_MODE

multiplicity: *single (static)*
type: *integer*

Daylight saving time beginning week

Remarks
- The property is enabled since protocol version *003*.

SYSM_DST2_MODE

multiplicity: *single (static)*
type: *integer*

Daylight saving time ending week

Remarks
- The property is enabled since protocol version *003*.

SYSM_UTC_OFFSET

multiplicity: *single (static)*
type: *integer*

SYSM_UTC_OFFSET

Remarks
- The property is enabled since protocol version *003*.

customer

multiplicity: *single (static)*
type: *integer*

CUSTOMER ID

Remarks
- The property is enabled since protocol version *002*.

region

multiplicity: *single (static)*
type: *integer*

SALES REGION

Remarks
- The property is enabled since protocol version *002*.

sysPanelLang

multiplicity: *single (static)*
type: *integer*

SYSM_PANELLANG

value	symbol
208	ENGLISH UK
144	GERMAN
9	DANISH
68	ITALIAN
224	SPANISH
136	FRENCH
64	DUTCH
76	NORWEGIAN-BOKMAL
80	PORTUGUESE
88	SWEDISH
22	POLISH
65	DUTCH BELG
137	FRENCH BELG
25	SLOVAK
13	FINNISH
8	CZECH
28	TURKISH
6	CATALAN
16	HUNGARIAN

begin.changeSessionKey

direction: *output*

This message is used to initiate procedure of changing encryption key used during the communication session with the panel.

The panel response for the message is `return.changeSessionKey`.

Both messages the call and the response consists of binary data (16 bytes each) that is used to creates new encryption key for the current session.

The new encryption key shall be changed as the successfull response (`return.void`) is received for `end.changeSessionKey` message that ends the procedure.

The length of the new encryption key must be the same as the current one. To create the key use first bytes from the call as first half of the key and the first bytest from the response as the second half of the key.

Remarks

- The message is available since protocol version *008*.
- The procedure of changing encryption key is valid only when the communication channel is encrypted.
- The procedure of changing encryption key is required at the beginning of the encrypted communication.

See also

- `end.changeSessionKey`
- `return.changeSessionKey`

data

multiplicity: *single (static)*
type: *string*

Data to create the new encryption key (16 bytes).

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

end.changeSessionKey

direction: *output*

This message is used to end procedure of changing encryption key used during the communication session with the panel.

After succesfull response (`return.void`) the new key should be used in the communication.

For further details see in `begin.changeSessionKey` message.

Remarks

- The message is available since protocol version *008*.

See also

- `begin.changeSessionKey`

return.changeSessionKey

direction: *input*

This message is used to continue procedure of changing encryption key used during the communication session with the panel.

For further details see in `begin.changeSessionKey` message.

Remarks

- The message is available since protocol version *008*.

See also

- `begin.changeSessionKey`

data

multiplicity: *single (static)*
type: *string*

Data to create the new encryption key (16 bytes).

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

device.getConnect

direction: *output*

This is the outgoing call for *device.getConnect* method. Remote user login.

If panel protocol version is between *001* and *010* all the actions flags are treated as required. Fault response means failed logon with the reason included in the fault message.

Since panel protocol version *011* all the action flags are treated as optional and the function is successful if the operator has riths to at least one of the selected action. To see whether the action is allowed after "successfull" logon use get.UserInfo and return.UserInfo pair of messages. If there is detected lack of the "required" access flag treat the logon as "failed" and simply disconnect the session with device.disconnect message.

userAction_UPLOAD

multiplicity: *single (static)*
type: *integer*

User action UPLOAD. Purpose of connection.

value	symbol
0	false
1	true

userAction_DOWNLOAD

multiplicity: *single (static)*
type: *integer*

User action DOWNLOAD. Purpose of connection.

value	symbol
0	false
1	true

userAction_CTRL

multiplicity: *single (static)*
type: *integer*

User action CTRL. Purpose of connection.

value	symbol
0	false
1	true

userAction_MONITOR

multiplicity: *single (static)*
type: *integer*

User action MONITOR. Purpose of connection.

value	symbol
0	false
1	true

userAction_DIAG

multiplicity: *single (static)*
type: *integer*

User action MONITOR. Purpose of connection.

value	symbol
0	false
1	true

userAction_LOGREAD

multiplicity: *single (static)*
type: *integer*

User action MONITOR. Purpose of connection.

value	symbol
0	false
1	true

userPIN

multiplicity: *single (static)*
type: *string*

User Pin.

device.disconnect

direction: *output*

This is the outgoing call for "device.disconnect" method.

is.Alive

direction: *output*

This message is usefull to *ping* the connected panel to check whether the conection is running.

The response to this command is `return.void`.

open.LOG

direction: *output*

This message opens events log in the panel for reading.

To close the log reading session use `close.LOG` request.

To read events from the log use `select.getLOG` request.

close.LOG

direction: *output*

This message ends session of reading events log from the panel.

See also

- open.LOG

select.getLOG

direction: *output*

This message reads one event from the event log.

The response message is `return.getLOG`.

See also

- `open.LOG`

logReadingDirection

multiplicity: *single (static)*
type: *integer*

Log reading parameter.

value	symbol
0	LOG_LATEST_EV
1	LOG_OLDEST_EV
2	LOG_NEWER_EV
3	LOG_OLDER_EV

return.getLOG

direction: *input*

This is the incoming message for "select.getLOG" method.

The available data in the message are the same as in `msg.MONITOR`.

See also
- `select.getLOG`
- `msg.MONITOR`

timeStamp

multiplicity: *single (static)*
type: *datetime*

The event timestamp.

Remarks
- The timestamp value is rounded to seconds precision.
- It is assumed that the value uses Coordinated Universal Time format aka *UTC*.
- Leap seconds available in *UTC* are not supported.

format: `date+time`

unique_id

multiplicity: *single (static)*
type: *integer*

Unique identifier of the event in the log.

logType

multiplicity: *single (static)*
type: *integer*

Log type

value	symbol
0	MANDATORY
1	NON_MANDATORY
2	INSTALLER
3	ACCESS

event_ID

multiplicity: *single (static)*
type: *integer*

Event type identifier.

value	symbol
0	ev_NULL
1	ev_AN
2	ev_ARZN
3	ev_ARDGP
4	ev_AS
5	ev_ATZN
6	ev_ATDGP
7	ev_BA
8	ev_BB
9	ev_BCUSER
10	ev_BCKEY
11	ev_BCPC
12	ev_BJ
13	ev_BR
14	ev_BT
15	ev_BU
16	ev_BV
17	ev_BW
18	ev_BZ
19	ev_CFUSER
20	ev_CFKEY
21	ev_CFPC
22	ev_CGUSER
23	ev_CGKEY
24	ev_CGPC
25	ev_CLUSER
26	ev_CLKEY
27	ev_CLPC
28	ev_EEUSER
29	ev_EESYST
30	ev_ERDGP
31	ev_ERDGPC
32	ev_ERDGPF
33	ev_ERRAS
34	ev_ERRASC
35	ev_ETDGP
36	ev_ETDGPC
37	ev_ETDGPF
38	ev_ETRAS

value	symbol
39	ev_ETRASC
40	ev_FA
41	ev_FB
42	ev_FJ
43	ev_FR
44	ev_FT
45	ev_FU
46	ev_FW
47	ev_HA
48	ev_HR
49	ev_JP
50	ev_JR
51	ev_JT
52	ev_LB
53	ev_LS
54	ev_LR
55	ev_LT
56	ev_MA
57	ev_MB
58	ev_MJ
59	ev_MR
60	ev_MU
61	ev_MS
62	ev_OPUSER
63	ev_OPKEY
64	ev_OPPC
65	ev_ORUSER
66	ev_ORKEY
67	ev_ORPC
68	ev_PA
69	ev_PB
70	ev_PJ
71	ev_PR
72	ev_PT
73	ev_PU
74	ev_RB
75	ev_RR
76	ev_RRDGP
77	ev_RRRAS
78	ev_RS
79	ev_RU
80	ev_RX
81	ev_TA
82	ev_TADGP
83	ev_TADGPS

value	symbol
84	ev_TARAS
85	ev_TB
86	ev_TR
87	ev_TRDGP
88	ev_TRDGPS
89	ev_TRRAS
90	ev_TU
91	ev_UBDGP
92	ev_UBRAS
93	ev_UBZN
94	ev_UUDGP
95	ev_UURAS
96	ev_UUZN
97	ev_WP
98	ev_WF
99	ev_XH
100	ev_XQ
101	ev_XR
102	ev_XT
103	ev_YC
104	ev_YK
105	ev_YR
106	ev_YS
107	ev_YT
108	ev_ZA
109	ev_ZB
110	ev_ZJ
111	ev_ZR
112	ev_ZS
113	ev_ZU
114	ev_OUTA
115	ev_OUTR
116	ev_USRG
117	ev_USRD
118	ev_EROK
119	ev_INTEG
120	ev_WDG
121	ev_CHKSUM
122	ev_USRADD
123	ev_USRDEL
124	ev_USRPIN
125	ev_USRTAMP
126	ev_ZNNTST
127	ev_BJS
128	ev_BZS

value	symbol
129	ev_INSTRST
130	ev_TT
131	ev_KBTR
132	ev_YTL
133	ev_OVRD
134	ev_YA
135	ev_YH
136	ev_NC
137	ev_NR
138	ev_LTETH
139	ev_LTIP
140	ev_LRIP
141	ev_JTSYS
142	ev_YSSYS
143	ev_YKSYS
144	ev_RP
145	ev_SMSUN
146	ev_SMSLIM
147	ev_YKMI
148	ev_LTGPRS
149	ev_LTIPGPRS
150	ev_LRIPGPRS
151	ev_CFRAS
152	ev_CGRAS
153	ev_CP
154	ev_OA
155	ev_OT
156	ev_OK
157	ev_AASTOPU
158	ev_AASTOPS
159	ev_PARAS
160	ev_PAUSR
161	ev_IA
162	ev_CA
163	ev_CR
164	ev_GA
165	ev_GR
166	ev_GB
167	ev_GU
168	ev_GS
169	ev_GJ
170	ev_KA
171	ev_KR
172	ev_KB
173	ev_KU

value	symbol
174	ev_KS
175	ev_KJ
176	ev_WA
177	ev_WR
178	ev_WB
179	ev_WU
180	ev_WS
181	ev_WJ
182	ev_CLRAS
183	ev_HV
184	ev_HW
185	ev_EEZONE
186	ev_UA
187	ev_UR
188	ev_FARAS
189	ev_MARAS
190	ev_FRRAS
191	ev_MRRAS
192	ev_PRRAS
193	ev_PRUSR
194	ev_PICRQ
195	ev_PICTKN
196	ev_PICSNT
197	ev_OHKM
198	ev_PICGPRSLIM
199	ev_PICETHLIM
200	ev_PICLIM
201	ev_PEBMEMFULL
202	ev_PEBMEMWRC90P
203	ev_GETIMGFAIL
204	ev_PICDEL
205	ev_CAMMODEFAIL
206	ev_TS
207	ev_TE
208	ev_LU
209	ev_BRAM
210	ev_BTAM
211	ev_BRFT
212	ev_BTFT
213	ev_PICMMSLIM
214	ev_TSDGP
215	ev_TEDGP
216	ev_TSSYS
217	ev_TESYS
218	ev_YPDGP

value	symbol
219	ev_YQDGP

event_source

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Source of the event (Keypad, Expander, PANEL, USER ... other). The detailed index (if available) is stored in the `source_ID` property.

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

source_ID

multiplicity: *single (static)*
type: *integer*

Index of the event source described in the `event_source` property.

Area

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Area index.

value	symbol
1	Area1
2	Area2
3	Area3
4	Area4
5	Area5
6	Area6
7	Area7
8	Area8

detailsFuse

multiplicity: *single (static)*
type: *integer*

Additional details that consists fuse ID.

Remarks

- The property is enabled if `event_ID` property equals `ev_ERDGPF` or `ev_ETDGPF`.

value	symbol
1	Internal_siren
2	External_siren
3	Battery
4	Communication
5	Aux_power_supply
6	DGP

detailsUserID

multiplicity: *single (static)*
type: *integer*

Additional details that consists user ID.

Remarks

- The property is enabled if `event_ID` property equals `ev_USRADD` or `ev_USRDEL` or `ev_PICDEL`.

detailsSiren

multiplicity: *single (static)*
type: *integer*

Additional details that consists siren ID.

Remarks

- The property is enabled if `event_ID` property equals `ev_TADGPS` or `ev_TRDGPS`.

value	symbol
1	Internal_siren
2	External_siren
3	On_board_siren_tamper_input

detailsTamper

multiplicity: *single (static)*
type: *integer*

Additional details that consists tamper ID.

Remarks

- The property is enabled if `event_ID` property equals `ev_TADGP` or `ev_TRDGP`.

value	symbol
1	Panel_tamper
2	Siren_tamper

detailsCS

multiplicity: *single (static)*
type: *integer*

Additional details that consists central station ID.

Remarks

- The property is enabled if `event_ID` property equals `ev_RX` or `ev_RP`.

min	max
1	16

detailsTimestamp

multiplicity: *single (static)*
type: *datetime*

Additional details that consists of timestamp value.

Remarks

- The property is enabled if `event_ID` property equals `ev_JT`.
- The timestamp value is rounded to seconds precision.
- It is assumed that the value uses Coordinated Universal Time format aka *UTC*.

- Leap seconds available in *UTC* are not supported.

format: `date+time`

`detailsShockGross`

multiplicity: *single (static)*
type: *integer*

Gross level activation value

Remarks
- The property is available since protocol version *002*.
- The property can be enabled if `event_ID` property equals `ev_BA`.

`detailsShockPulse`

multiplicity: *single (static)*
type: *integer*

Pulse level activation value

Remarks
- The property is available since protocol version *002*.
- The property can be enabled if `event_ID` property equals `ev_BA`.

`detailsSmsCnt`

multiplicity: *single (static)*
type: *integer*

SMS counter identifier

Remarks
- The property is available since protocol version *011*.
- The property can be enabled if `event_ID` property equals `ev_SMSLIM`.

value	symbol
1	Reporting
2	Forwarding

`sessionType`

multiplicity: *single (static)*
type: *integer*

Additional details - user logon session type.

Remarks
- The property is available since protocol version *015*.

- The property is enabled if `event_ID` property equals `ev_USRD` or `ev_USRG`.

value	symbol
1	LOCAL
2	REMOTE
3	KEYFOB
4	CARD
5	SMS

detailsRAS

multiplicity: *single (static)*
type: *integer*

Additional details - user logon RAS index.

Remarks

- The property is available since protocol version *015*.
- The property is enabled if `event_ID` property equals `ev_USRD` or `ev_USRG` and `sessionType` property equals `CARD` or `LOCAL`.

min	max
1	8

detailsPictureId

multiplicity: *single (static)*
type: *integer*

Additional details that consists of picture id value.

detailsPictureSource

multiplicity: *single (static)*
type: *integer*

Additional details that consists of picture source value.

value	symbol
0	ZONE
1	FILTER
2	TEST_CALL
3	WALKTEST
4	SMS
5	CC
6	SIA

detailsPictureSourceId

multiplicity: *single (static)*
type: *integer*

Additional details that consists of picture source id value.

detailsCameraMode

multiplicity: *single (static)*
type: *integer*

Additional details that contain the requested camera mode.

Remarks

- The property is enabled if `event_ID` property equals `ev_CAMMODEFAIL`.

value	symbol
1	DAY
2	NIGHT
4	WALKTEST
8	RANGETEST
16	LEGACY
32	ISOLATE
33	DAY_ISOLATE
34	NIGHT_ISOLATE
36	WALKTEST_ISOLATE
40	RANGETEST_ISOLATE
48	LEGACY_ISOLATE

event_text

multiplicity: *single (static)*
type: *string*

Extra info text attached to the event.

Remarks

- The property is available since protocol version *014*.
- The property is unavailable if `userCARD` property enabled.

userCARD

multiplicity: *single (static)*
type: *string*

User card data.

Remarks

- The only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.
- The property is available since protocol version *015*.
- The property is enabled if `event_ID` property equals `ev_USRD` and `sessionType` property equals `CARD`.

start.MONITOR

direction: *output*

This message is used to start the unsolicited transmissions of log events from panel to PC.

The response is `return.bool`.

Remarks

- The message is available since protocol version *011*.

stop.MONITOR

direction: *output*

This message is used to stop the unsolicited transmissions of log events from panel to PC.

The response is `return.bool`.

Remarks

- The message is available since protocol version *011*.

pause.MONITOR

direction: *output*

This message is used to pause the unsolicited transmissions of log events from panel to PC, and preserve log read position.

The response is `return.bool`.

Remarks

- The message is available since protocol version *011*.

msg.MONITOR

direction: *input*

This is the message for unsolicited event transmission from panel to Management Software.

The received event need to be confirmed (`return true`) by the proper handler (`IEventListener`) in the application.

The available data in the message are the same as in `return.getLOG`.

See also
- `return.getLOG`

timeStamp

multiplicity: *single (static)*
type: *datetime*

The event timestamp.

Remarks
- The timestamp value is rounded to seconds precision.
- It is assumed that the value uses Coordinated Universal Time format aka *UTC*.
- Leap seconds available in *UTC* are not supported.

format: `date+time`

unique_id

multiplicity: *single (static)*
type: *integer*

Unique identifier of the event in the log.

logType

multiplicity: *single (static)*
type: *integer*

Log type

value	symbol
0	MANDATORY
1	NON_MANDATORY

value	symbol
2	INSTALLER
3	ACCESS

event_ID

multiplicity: *single (static)*
type: *integer*

Event type identifier.

value	symbol
0	ev_NULL
1	ev_AN
2	ev_ARZN
3	ev_ARDGP
4	ev_AS
5	ev_ATZN
6	ev_ATDGP
7	ev_BA
8	ev_BB
9	ev_BCUSER
10	ev_BCKEY
11	ev_BCPC
12	ev_BJ
13	ev_BR
14	ev_BT
15	ev_BU
16	ev_BV
17	ev_BW
18	ev_BZ
19	ev_CFUSER
20	ev_CFKEY
21	ev_CFPC
22	ev_CGUSER
23	ev_CGKEY
24	ev_CGPC
25	ev_CLUSER
26	ev_CLKEY
27	ev_CLPC
28	ev_EEUSER
29	ev_EESYST
30	ev_ERDGP
31	ev_ERDGPC
32	ev_ERDGPF
33	ev_ERRAS
34	ev_ERRASC

value	symbol
35	ev_ETDGP
36	ev_ETDGPC
37	ev_ETDGPF
38	ev_ETRAS
39	ev_ETRASC
40	ev_FA
41	ev_FB
42	ev_FJ
43	ev_FR
44	ev_FT
45	ev_FU
46	ev_FW
47	ev_HA
48	ev_HR
49	ev_JP
50	ev_JR
51	ev_JT
52	ev_LB
53	ev_LS
54	ev_LR
55	ev_LT
56	ev_MA
57	ev_MB
58	ev_MJ
59	ev_MR
60	ev_MU
61	ev_MS
62	ev_OPUSER
63	ev_OPKEY
64	ev_OPPC
65	ev_ORUSER
66	ev_ORKEY
67	ev_ORPC
68	ev_PA
69	ev_PB
70	ev_PJ
71	ev_PR
72	ev_PT
73	ev_PU
74	ev_RB
75	ev_RR
76	ev_RRDGP
77	ev_RRRAS
78	ev_RS
79	ev_RU

value	symbol
80	ev_RX
81	ev_TA
82	ev_TADGP
83	ev_TADGPS
84	ev_TARAS
85	ev_TB
86	ev_TR
87	ev_TRDGP
88	ev_TRDGPS
89	ev_TRRAS
90	ev_TU
91	ev_UBDGP
92	ev_UBRAS
93	ev_UBZN
94	ev_UUDGP
95	ev_UURAS
96	ev_UUZN
97	ev_WP
98	ev_WF
99	ev_XH
100	ev_XQ
101	ev_XR
102	ev_XT
103	ev_YC
104	ev_YK
105	ev_YR
106	ev_YS
107	ev_YT
108	ev_ZA
109	ev_ZB
110	ev_ZJ
111	ev_ZR
112	ev_ZS
113	ev_ZU
114	ev_OUTA
115	ev_OUTR
116	ev_USRG
117	ev_USRD
118	ev_EROK
119	ev_INTEG
120	ev_WDG
121	ev_CHKSUM
122	ev_USRADD
123	ev_USRDEL
124	ev_USRPIN

value	symbol
125	ev_USRTAMP
126	ev_ZNNTST
127	ev_BJS
128	ev_BZS
129	ev_INSTRST
130	ev_TT
131	ev_KBTR
132	ev_YTL
133	ev_OVRD
134	ev_YA
135	ev_YH
136	ev_NC
137	ev_NR
138	ev_LTETH
139	ev_LTIP
140	ev_LRIP
141	ev_JTSYS
142	ev_YSSYS
143	ev_YKSYS
144	ev_RP
145	ev_SMSUN
146	ev_SMSLIM
147	ev_YKMI
148	ev_LTGPRS
149	ev_LTIPGPRS
150	ev_LRIPGPRS
151	ev_CFRAS
152	ev_CGRAS
153	ev_CP
154	ev_OA
155	ev_OT
156	ev_OK
157	ev_AASTOPU
158	ev_AASTOPS
159	ev_PARAS
160	ev_PAUSR
161	ev_IA
162	ev_CA
163	ev_CR
164	ev_GA
165	ev_GR
166	ev_GB
167	ev_GU
168	ev_GS
169	ev_GJ

value	symbol
170	ev_KA
171	ev_KR
172	ev_KB
173	ev_KU
174	ev_KS
175	ev_KJ
176	ev_WA
177	ev_WR
178	ev_WB
179	ev_WU
180	ev_WS
181	ev_WJ
182	ev_CLRAS
183	ev_HV
184	ev_HW
185	ev_EEZONE
186	ev_UA
187	ev_UR
188	ev_FARAS
189	ev_MARAS
190	ev_FRRAS
191	ev_MRRAS
192	ev_PRRAS
193	ev_PRUSR
194	ev_PICRQ
195	ev_PICTKN
196	ev_PICSNT
197	ev_OHKM
198	ev_PICGPRSLIM
199	ev_PICETHLIM
200	ev_PICLIM
201	ev_PEBMEMFULL
202	ev_PEBMEMWRC90P
203	ev_GETIMGFAIL
204	ev_PICDEL
205	ev_CAMMODEFAIL
206	ev_TS
207	ev_TE
208	ev_LU
209	ev_BRAM
210	ev_BTAM
211	ev_BRFT
212	ev_BTFT
213	ev_PICMMSLIM
214	ev_TSDGP

value	symbol
215	ev_TEDGP
216	ev_TSSYS
217	ev_TESYS
218	ev_YPDGP
219	ev_YQDGP

event_source

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Source of the event (Keypad, Expander, PANEL, USER ... other). The detailed index (if available) is stored in the `source_ID` property.

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

source_ID

multiplicity: *single (static)*
type: *integer*

Index of the event source described in the `event_source` property.

Area

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Area index.

value	symbol
1	Area1
2	Area2
3	Area3
4	Area4
5	Area5
6	Area6
7	Area7
8	Area8

detailsFuse

multiplicity: *single (static)*
type: *integer*

Additional details that consists fuse ID.

Remarks

- The property is enabled if `event_ID` property equals `ev_ERDGPF` or `ev_ETDGPF`.

value	symbol
1	Internal_siren
2	External_siren
3	Battery
4	Communication
5	Aux_power_supply
6	DGP

detailsUserID

multiplicity: *single (static)*
type: *integer*

Additional details that consists user ID.

Remarks

- The property is enabled if `event_ID` property equals `ev_USRADD` or `ev_USRDEL` or `ev_PICDEL`.

detailsSiren

multiplicity: *single (static)*
type: *integer*

Additional details that consists siren ID.

Remarks
- The property is enabled if `event_ID` property equals `ev_TADGPS` or `ev_TRDGPS`.

value	symbol
1	Internal_siren
2	External_siren
3	On_board_siren_tamper_input

detailsTamper

multiplicity: *single (static)*
type: *integer*

Additional details that consists tamper ID.

Remarks
- The property is enabled if `event_ID` property equals `ev_TADGP` or `ev_TRDGP`.

value	symbol
1	Panel_tamper
2	Siren_tamper

detailsCS

multiplicity: *single (static)*
type: *integer*

Additional details that consists central station ID.

Remarks
- The property is enabled if `event_ID` property equals `ev_RX` or `ev_RP`.

min	max
1	16

detailsTimestamp

multiplicity: *single (static)*
type: *datetime*

Additional details that consists of timestamp value.

Remarks
- The property is enabled if `event_ID` property equals `ev_JT`.
- The timestamp value is rounded to seconds precision.
- It is assumed that the value uses Coordinated Universal Time format aka *UTC*.
- Leap seconds available in *UTC* are not supported.

format: `date+time`

detailsShockGross

multiplicity: *single (static)*
type: *integer*

Gross level activation value

Remarks
- The property is available since protocol version *002*.
- The property can be enabled if `event_ID` property equals `ev_BA`.

detailsShockPulse

multiplicity: *single (static)*
type: *integer*

Pulse level activation value

Remarks
- The property is available since protocol version *002*.
- The property can be enabled if `event_ID` property equals `ev_BA`.

detailsSmsCnt

multiplicity: *single (static)*
type: *integer*

SMS counter identifier

Remarks
- The property is available since protocol version *011*.
- The property can be enabled if `event_ID` property equals `ev_SMSLIM`.

value	symbol
1	Reporting
2	Forwarding

sessionType

multiplicity: *single (static)*
type: *integer*

Additional details - user logon session type.

Remarks
- The property is available since protocol version *015*.
- The property is enabled if `event_ID` property equals `ev_USRD` or `ev_USRG`.

value	symbol
1	LOCAL
2	REMOTE
3	KEYFOB
4	CARD
5	SMS

detailsRAS

multiplicity: *single (static)*
type: *integer*

Additional details - user logon RAS index.

Remarks
- The property is available since protocol version *015*.
- The property is enabled if `event_ID` property equals `ev_USRD` or `ev_USRG` and `sessionType` property equals `CARD` or `LOCAL`.

min	max
1	8

detailsPictureId

multiplicity: *single (static)*
type: *integer*

Additional details that consists of picture id value.

detailsPictureSource

multiplicity: *single (static)*
type: *integer*

Additional details that consists of picture source value.

value	symbol
0	ZONE
1	FILTER

value	symbol
2	TEST_CALL
3	WALKTEST
4	SMS
5	CC
6	SIA

detailsPictureSourceId

multiplicity: *single (static)*
type: *integer*

Additional details that consists of picture source id value.

detailsCameraMode

multiplicity: *single (static)*
type: *integer*

Additional details that contain the requested camera mode.

Remarks

- The property is enabled if `event_ID` property equals `ev_CAMMODEFAIL`.

value	symbol
1	DAY
2	NIGHT
4	WALKTEST
8	RANGETEST
16	LEGACY
32	ISOLATE
33	DAY_ISOLATE
34	NIGHT_ISOLATE
36	WALKTEST_ISOLATE
40	RANGETEST_ISOLATE
48	LEGACY_ISOLATE

event_text

multiplicity: *single (static)*
type: *string*

Extra info text attached to the event.

Remarks

- The property is available since protocol version *014*.
- The property is unavailable if `userCARD` property enabled.

userCARD

multiplicity: *single (static)*
type: *string*

User card data.

Remarks

- The only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.
- The property is available since protocol version *015*.
- The property is enabled if `event_ID` property equals `ev_USRD` and `sessionType` property equals `CARD`.

msgCOS.ALL

direction: *input*

This message is sent asynchronously from panel to Management Software, when *DIAGNOSE* access level is enabled, and contains *change of state* flags for all system objects.

The received event need to be confirmed (`return true`) by the proper handler (`IEventListener`) in the application.

Remarks

- The message is available since protocol version *011*.

APPOBJ_ZN

multiplicity: *single (static)*
type: *boolean*

ZONE flags changed.

To find all zones with changed state `getCOS.ZONE` request need to be used.

To read particular zone state `getSTAT.ZONE` request need to be used.

APPOBJ_AREA

multiplicity: *single (static)*
type: *boolean*

AREA flags changed.

To find all areas with changed state `getCOS.AREA` request need to be used.

To read particular area state `getSTAT.AREA` request need to be used.

APPOBJ_RAS

multiplicity: *single (static)*
type: *boolean*

Keypas flags changed.

To find all keypads with changed state `getCOS.RAS` request need to be used.

To read particular keypad state `getSTAT.RAS` request need to be used.

APPOBJ_DGP

multiplicity: *single (static)*
type: *boolean*

Expander flags changed.

To find all expanders with changed state `getCOS.DGP` request need to be used.

To read particular expander state `getSTAT.DGP` request need to be used.

APPOBJ_DGP0

multiplicity: *single (static)*
type: *boolean*

DGP0 flags changed.

To read current panel state `getSTAT.DGP0` request need to be used.

APPOBJ_USER

multiplicity: *single (static)*
type: *boolean*

USER flags changed.

To find all users with changed state `getCOS.USER` request need to be used.

To read particular user state `getSTAT.USER` request need to be used.

APPOBJ_OUT

multiplicity: *single (static)*
type: *boolean*

OUTPUT flags changed.

To find all outputs with changed state `getCOS.OUT` request need to be used.

To read particular output state `getSTAT.OUT` request need to be used.

APPOBJ_CEVFLT

multiplicity: *single (static)*
type: *boolean*

CEVFILTER flags changed.

To find all condition filters with changed state `getCOS.FILTER` request need to be used.

To read particular condition filter state `getSTAT.FILTER` request need to be used.

APPOBJ_UG

multiplicity: *single (static)*
type: *boolean*

USERGROUP flags changed.

To find all user groups with changed state `getCOS.UG` request need to be used.

To read particular user group state `getSTAT.UG` request need to be used.

APPOBJ_SYS

multiplicity: *single (static)*
type: *boolean*

SYSTEM flags changed.

To read current system state `getSTAT.SYS` request need to be used.

APPOBJ_CS

multiplicity: *single (static)*
type: *boolean*

CS flags changed.

To find all central stations with changed state `getCOS.CS` request need to be used.

To read particular central station state `getSTAT.CS` request need to be used.

APPOBJ_PCC

multiplicity: *single (static)*
type: *boolean*

PCCONN flags changed.

To find all PC connections with changed state `getCOS.PCC` request need to be used.

To read particular PC connection state `getSTAT.PCC` request need to be used.

APPOBJ_TRIGG

multiplicity: *single (static)*
type: *boolean*

TRIGGER flags changed.

To find all triggers with changed state `getCOS.TRIGG` request need to be used.

To read particular trigger state `getSTAT.TRIGG` request need to be used.

APPOBJ_SCHDL_EXC

multiplicity: *single (static)*
type: *boolean*

SCHDL EXC flags changed.

To find all scheduler exceptions with changed state `getCOS.EXCP` request need to be used.

To read particular scheduler exception state `getSTAT.EXCP` request need to be used.

Remarks
- The property is available since protocol version *015*.

APPOBJ_SCHDL_CAL

multiplicity: *single (static)*
type: *boolean*

SCHDL CAL flags changed.

To read current scheduler state `getSTAT.SCAL` request need to be used.

Remarks
- The property is available since protocol version *015*.

APPOBJ_FOB

multiplicity: *single (static)*
type: *boolean*

FOB flags changed.

To find all fobs with changed state `getCOS.FOB` request need to be used.

To read particular fob state `getSTAT.FOB` request need to be used.

Remarks

- The property is available since protocol version *018*.

APPOBJ_CAMERA

multiplicity: *single (static)*
type: *boolean*

CAMERA flags changed.

To find all cameras with changed state `getCOS.CAMERA` request need to be used.

To read particular fob state `getSTAT.CAMERA` request need to be used.

Remarks

- The property is available since protocol version *022*.

msgCOS.CAM_RANGETST

direction: *input*

This message is sent asynchronously from panel to Management Software, when *DIAGNOSE* access level is enabled, and contains *change of state* data for camera range test.

The received event need to be confirmed (`return true`) by the proper handler (`IEventListener`) in the application.

Remarks

- The message is available since protocol version *021*.

camera

multiplicity: *single (static)*
type: *integer*

status

multiplicity: *single (static)*
type: *integer*

Status indicates that test passed (true) or not (false). In the case of false, remaining fields are not valid

value	symbol
0	false
1	true

zone

multiplicity: *single (static)*
type: *integer*

LDR_RSSI

multiplicity: *single (static)*
type: *integer*

LDR_L_avgRSSI

multiplicity: *single (static)*
type: *integer*

LDR_R_avgRSSI

multiplicity: *single (static)*
type: *integer*

LDR_L_packetCount

multiplicity: *single (static)*
type: *integer*

LDR_R_packetCount

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel1

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel2

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel3

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel4

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel5

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel6

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel7

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel8

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel9

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel10

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel11

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel12

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel13

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel14

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel15

multiplicity: *single (static)*
type: *integer*

HDR_QIchannel16

multiplicity: *single (static)*
type: *integer*

HDR_QImax

multiplicity: *single (static)*
type: *integer*

HDR_QImin

multiplicity: *single (static)*
type: *integer*

HDR_QIavg

multiplicity: *single (static)*
type: *integer*

msgCOS.SYS_INV_WALKTST_REP

direction: *input*

This message is sent asynchronously from panel to Management Software, when *DIAGNOSE* access level is enabled, and contains *change of state* data for inverted walk test.

The received event need to be confirmed (`return true`) by the proper handler (`IEventListener`) in the application.

Remarks

- The message is available since protocol version *023*.

zone

multiplicity: *single (static)*
type: *integer*

Zone index.

status

multiplicity: *single (static)*
type: *integer*

Status indicates that maximum time for zone activation expired.

value	symbol
0	false
1	true

time

multiplicity: *single (static)*
type: *integer*

The time in seconds since the zone was recently activated.

zone_name

multiplicity: *single (static)*
type: *string*

Zone name.

get.timedate

direction: *output*

This message is used to get the current time and date from the panel.

The response is `return.timedate`.

Remarks

- The message is available since protocol version *011*.

return.timedate

direction: *input*

This message returns the current time and date from the panel.

See also
- get.timedate

timeStamp

multiplicity: *single (static)*
type: *datetime*

The timestamp value with current time and date in the panel.

Remarks
- It is assumed that the value uses Coordinated Universal Time format aka *UTC*.
- Leap seconds available in *UTC* are not supported.

format: date+time

get.privileges

direction: *output*

This message is used to get the current session privileges for the specified area, or system privileges (index 0).

The response is `return.privileges`.

Remarks

- The message is available since protocol version *011*.

areaID

multiplicity: *single (static)*
type: *integer*

Area identifier (1-...) or 0 (system privileges).

return.privileges

direction: *input*

This message returns the information about the privileges for the specified area, or system privileges (index 0).

See also

- get.privileges

areaID

multiplicity: *single (static)*
type: *integer*

Area identifier (1-...) or 0 (system privileges).

UGP_FULLSET

multiplicity: *single (static)*
type: *boolean*

User can Set the area in Full mode.

This is area privilege.

UGP_PARTSET

multiplicity: *single (static)*
type: *boolean*

User can Set the area in Part mode.

This is area privilege.

UGP_UNSET

multiplicity: *single (static)*
type: *boolean*

User can unset the area.

This is area privilege.

UGP_INHIBIT

multiplicity: *single (static)*
type: *boolean*

User can inhibit zones.

This is area privilege.

UGP_ISOLATE

multiplicity: *single (static)*
type: *boolean*

User can isolate zones.

This is area privilege.

UGP_TIMEDATE

multiplicity: *single (static)*
type: *boolean*

User can change system Time and Date.

This is system privilege.

UGP_CUSER

multiplicity: *single (static)*
type: *integer*

User creation mode.

This is system privilege.

value	symbol
0	UGP_NCUSER
1	UGP_RCUSER
2	UGP_FCUSER

UGP_FSET

multiplicity: *single (static)*
type: *boolean*

Forced Set areas is available.

This is area privilege.

UGP_CHGPIN

multiplicity: *single (static)*
type: *boolean*

User can change his PIN.

This is system privilege.

UGP_WALK

multiplicity: *single (static)*
type: *boolean*

Walktest function is available.

This is area privilege.

UGP_ENGRESET

multiplicity: *single (static)*
type: *boolean*

User can perform Engineer Reset.

This is system privilege.

UGP_DURESS

multiplicity: *single (static)*
type: *boolean*

User can use duress code.

This is system privilege.

UGP_TESTREP

multiplicity: *single (static)*
type: *boolean*

User can perform CS tests.

This is system privilege.

UGP_COMM

multiplicity: *single (static)*
type: *boolean*

User can access the panel via remote.

This is system privilege.

UGP_CLEANER

multiplicity: *single (static)*
type: *boolean*

Do not uninhibit zones after area Unset.

This is system privilege.

UGP_AREALIST

multiplicity: *single (static)*
type: *boolean*

User can list areas.

This is system privilege.

UGP_MENUACC

multiplicity: *single (static)*
type: *boolean*

User can access menu.

This is system privilege.

UGP_INSTACC

multiplicity: *single (static)*
type: *boolean*

Allow Installer Access.

This is system privilege.

UGP_VSTOP

multiplicity: *single (static)*
type: *boolean*

Stop Voice reporting.

This is area privilege.

UGP_LOGSACC

multiplicity: *single (static)*
type: *boolean*

Allow Log Access.

This is area privilege.

UGP_CARDPIN

multiplicity: *single (static)*
type: *integer*

Authentication type (pin/card).

This is system privilege.

value	symbol
0	Card or PIN
1	PIN only
2	Card only

UGP_SUPERVISOR

multiplicity: *single (static)*
type: *integer*

User type (supervisor/installer/guard).

This is system privilege.

value	symbol
0	Normal User
1	Supervisor
2	Installer
3	Guard

get.UserInfo

direction: *output*

This message is used to get the current User information.

The response is `return.UserInfo`.

Remarks

- The message is available since protocol version *011*.

return.UserInfo

direction: *input*

This message returns the information about the logged User.

See also
- get.UserInfo

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userName

multiplicity: *single (static)*
type: *string*

User name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

userLang

multiplicity: *single (static)*
type: *integer*

User language.

value	symbol
208	ENGLISH UK
144	GERMAN
9	DANISH
68	ITALIAN
224	SPANISH
136	FRENCH
64	DUTCH
76	NORWEGIAN-BOKMAL
80	PORTUGUESE
88	SWEDISH
22	POLISH
65	DUTCH BELG
137	FRENCH BELG
25	SLOVAK
13	FINNISH
8	CZECH
28	TURKISH
6	CATALAN
16	HUNGARIAN

SESSM_UPLOAD

multiplicity: *single (static)*
type: *boolean*

User can Upload configuration data from the panel.

SESSM_DOWNLOAD

multiplicity: *single (static)*
type: *boolean*

User can Download configuration data to the panel.

SESSM_CTRL

multiplicity: *single (static)*
type: *boolean*

User can execute Control functions.

SESSM_MONITOR

multiplicity: *single (static)*
type: *boolean*

User can receive monitoring events from the panel.

SESSM_DIAG

multiplicity: *single (static)*
type: *boolean*

User can read system status.

SESSM_LOGREAD

multiplicity: *single (static)*
type: *boolean*

User can read system Log.

get.liveEvents

direction: *output*

This message is used to get the currently present events from the specified class, for the specified area.

The response is `return.sysevent`.

Remarks

- The message is available since protocol version *011*.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

evCatFAULT

multiplicity: *single (static)*
type: *boolean*

The event belongs to FAULT category (active fault, except Mains).

evCatMAINS

multiplicity: *single (static)*
type: *boolean*

The event belongs to MAINS category (mains fault).

evCatACTZN

multiplicity: *single (static)*
type: *boolean*

The event belongs to ACTZN category (zones in active state).

evCatACT24H

multiplicity: *single (static)*
type: *boolean*

The event belongs to ACT24H category (zones in active state, with 24h option).

evCatACTLCD

multiplicity: *single (static)*
type: *boolean*

The event belongs to ACTLCD category (zones in active state, with LCD option).

evCatACTDEV

multiplicity: *single (static)*
type: *boolean*

The event belongs to ACTDEV category (devices in active state - tamper,offline ...).

evCatALARMS_NCNF

multiplicity: *single (static)*
type: *boolean*

The event belongs to ALARMS_NCNF category (not confirmed alarms).

evCatALARMS_CNF

multiplicity: *single (static)*
type: *boolean*

The event belongs to ALARMS_CNF category (confirmed alarms).

evCatFAULTS_NCNF

multiplicity: *single (static)*
type: *boolean*

The event belongs to FAULTS_NCNF category (not confirmed faults, recorded during set).

evCatFAULTS_CNF

multiplicity: *single (static)*
type: *boolean*

The event belongs to FAULTS_CNF category (confirmed faults, recorded during set).

evCatWALK_REQ

multiplicity: *single (static)*
type: *boolean*

The event belongs to WALK_REQ category (events required for walk test).

evCatWALK_OK

multiplicity: *single (static)*
type: *boolean*

The event belongs to WALK_OK category (events tested during walk test).

evCatSYSTEM

multiplicity: *single (static)*
type: *boolean*

The event belongs to SYSTEM category (system status events, e.g. eng.reset request).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

return.sysevent

direction: *input*

This message returns the event data (faults,alarms etc.)

See also

- get.liveEvents
- createCC.A_SET
- createCC.A_PARTSET
- createCC.A_PARTSET2
- fnCC.A_STATE_GET_INH
- fnCC.A_STATE_GET_UNINH
- fnCC.A_SET_GETFAULT
- fnCC.A_SET_GETACTIVE
- fnCC.A_SET_GETINHIB
- fnCC.A_CONFAL_GETALARM
- fnCC.A_WALKTST_GETLIST
- fnCC.A_WALKTST_GETEV
- fnCC.A_WALKTST_GETRES
- fnCC.A_UNSET_GETALARM
- fnCC.A_UNSET_GETFAULT

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

evCatFAULT

multiplicity: *single (static)*
type: *boolean*

The event belongs to FAULT category (active fault, except Mains).

evCatMAINS

multiplicity: *single (static)*
type: *boolean*

The event belongs to MAINS category (mains fault).

evCatACTZN

multiplicity: *single (static)*
type: *boolean*

The event belongs to ACTZN category (zones in active state).

evCatACT24H

multiplicity: *single (static)*
type: *boolean*

The event belongs to ACT24H category (zones in active state, with 24h option).

evCatACTLCD

multiplicity: *single (static)*
type: *boolean*

The event belongs to ACTLCD category (zones in active state, with LCD option).

evCatACTDEV

multiplicity: *single (static)*
type: *boolean*

The event belongs to ACTDEV category (devices in active state - tamper,offline ...).

evCatALARMS_NCNF

multiplicity: *single (static)*
type: *boolean*

The event belongs to ALARMS_NCNF category (not confirmed alarms).

evCatALARMS_CNF

multiplicity: *single (static)*
type: *boolean*

The event belongs to ALARMS_CNF category (confirmed alarms).

evCatFAULTS_NCNF

multiplicity: *single (static)*
type: *boolean*

The event belongs to FAULTS_NCNF category (not confirmed faults, recorded during set).

evCatFAULTS_CNF

multiplicity: *single (static)*
type: *boolean*

The event belongs to FAULTS_CNF category (confirmed faults, recorded during set).

evCatWALK_REQ

multiplicity: *single (static)*
type: *boolean*

The event belongs to WALK_REQ category (events required for walk test).

evCatWALK_OK

multiplicity: *single (static)*
type: *boolean*

The event belongs to WALK_OK category (events tested during walk test).

evCatSYSTEM

multiplicity: *single (static)*
type: *boolean*

The event belongs to SYSTEM category (system status events, e.g. eng.reset request).

eventUniqueID

multiplicity: *single (static)*
type: *integer*

Event unique identifier (16 bit).

classID

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Class identifier (16 bit).

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

objNum

multiplicity: *single (static)*
type: *integer*

Object number (16 bit).

eventTypeID

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Event type identifier (16 bit).

value	symbol
1	ARAC_ALARM
2	ARAC_TAMPER
3	ARAC_FIRE

value	symbol
4	ARAC_PANIC
5	ARAC_MEDICAL
6	ARAC_FIREDOOR
7	ARAC_AM
8	ARAC_DEVTAMPER
9	ARAC_SIRTAMPER
10	ARAC_DEVOFFLINE
11	ARAC_RFJAM
12	ARAC_LSUPALARM
13	ARAC_SOAK
14	ARAC_PALARMS
15	ARAC_EXITFAULT
16	ARAC_FAULT
17	ARAC_MAINSFAIL
18	ARAC_ZNBATTFAIL
19	ARAC_BATTFAIL
20	ARAC_BATTLOW
21	ARAC_TPATHFAULT
22	ARAC_ETHLINKFAULT
23	ARAC_IPLINKFAULT
24	ARAC_TECHNICAL
25	ARAC_FUSEFAULT
26	ARAC_SIRENFAULT
27	ARAC_RCVFAULT
28	ARAC_MIFAULT
29	ARAC_FTC
30	ARAC_ENGRES
31	ARAC_ZNACTIVE
32	ARAC_ZNFAULT
33	ARAC_ZNAM
34	ARAC_ZNTAMPER
35	ARAC_ZNINHIBIT
36	ARAC_ZNINHIBITFORCED
37	ARAC_ZNINHIBITONAB
38	ARAC_ZNSSUPFAULT
39	ARAC_ZNLSUP
40	ARAC_ISOLATE
41	ARAC_GPRSLINKFAULT
42	ARAC_IPGPRSLINKFAULT
43	ARAC_ZNDIRTY

getCOS.ZONE

direction: *output*

This message is used to get the COS flags for zones.

The response is `returnCOS.ZONE`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`

returnCOS.ZONE

direction: *input*

This message returns *change of state* bits for zones.

To read state of the changed zone `getSTAT.ZONE` request need to be used.

Remarks
- The message is available since protocol version *011*.
- For protocol between *011* and *018* the number of supported zones is 128.
- Since protocol version *019* the number of supported zones is 368.

See also
- `getCOS.ZONE`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```
the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```
the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.AREA

direction: *output*

This message is used to get the COS flags for areas.

The response is `returnCOS.AREA`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`

returnCOS.AREA

direction: *input*

This message returns *change of state* bits for areas.

To read state of the changed area `getSTAT.AREA` request need to be used.

Remarks

- The message is available since protocol version *011*.

See also

- `getCOS.AREA`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.RAS

direction: *output*

This message is used to get the COS flags for keypads.

The response is `returnCOS.RAS`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`

returnCOS.RAS

direction: *input*

This message returns *change of state* bits for keypads.

To read state of the changed keypad `getSTAT.RAS` request need to be used.

Remarks

- The message is available since protocol version *011*.

See also

- `getCOS.RAS`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.DGP

direction: *output*

This message is used to get the COS flags for expanders.

The response is `returnCOS.DGP`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`

returnCOS.DGP

direction: *input*

This message returns *change of state* bits for expanders.

To read state of the changed expander `getSTAT.DGP` request need to be used.

Remarks

- The message is available since protocol version *011*.

See also

- `getCOS.DGP`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.OUT

direction: *output*

This message is used to get the COS flags for OUTPUTs.

The response is `returnCOS.OUT`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`

returnCOS.OUT

direction: *input*

This message returns *change of state* bits for outputs.

To read state of the changed output `getSTAT.OUT` request need to be used.

Remarks

- The message is available since protocol version *011*.

See also

- `getCOS.OUT`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.FILTER

direction: *output*

This message is used to get the COS flags for FILTERs.

The response is `returnCOS.FILTER`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`

returnCOS.FILTER

direction: *input*

This message returns *change of state* bits for condition filters.

To read state of the changed condition filter `getSTAT.FILTER` request need to be used.

Remarks

- The message is available since protocol version *011*.

See also

- `getCOS.FILTER`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.PCC

direction: *output*

This message is used to get the COS flags for PC connections.

The response is `returnCOS.PCC`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`

returnCOS.PCC

direction: *input*

This message returns *change of state* bits for PC connections.

To read state of the changed PC connection `getSTAT.PCC` request need to be used.

Remarks

- The message is available since protocol version *011*.

See also

- `getCOS.PCC`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.CS

direction: *output*

This message is used to get the COS flags for central stations.

The response is `returnCOS.CS`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`

returnCOS.CS

direction: *input*

This message returns *change of state* bits for central stations.

To read state of the changed central station `getSTAT.CS` request need to be used.

Remarks

- The message is available since protocol version *011*.

See also

- `getCOS.CS`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.TRIGG

direction: *output*

This message is used to get the COS flags for TRIGGERs.

The response is `returnCOS.TRIGG`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`

returnCOS.TRIGG

direction: *input*

This message returns *change of state* bits for triggers.

To read state of the changed trigger `getSTAT.TRIGG` request need to be used.

Remarks

- The message is available since protocol version *011*.

See also

- `getCOS.TRIGG`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.USER

direction: *output*

This message is used to get the COS flags for USERs.

The response is `returnCOS.USER`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`

returnCOS.USER

direction: *input*

This message returns *change of state* bits for users.

To read state of the changed user `getSTAT.USER` request need to be used.

Remarks

- The message is available since protocol version *011*.

See also

- `getCOS.USER`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.UG

direction: *output*

This message is used to get the COS flags for USER GROUPs.

The response is `returnCOS.UG`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`

returnCOS.UG

direction: *input*

This message returns *change of state* bits for user groups.

To read state of the changed user group `getSTAT.UG` request need to be used.

Remarks

- The message is available since protocol version *011*.

See also

- `getCOS.UG`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.EXCP

direction: *output*

This message is used to get the COS flags for schedule exceptions.

The response is `returnCOS.EXCP`.

Remarks

- The message is available since protocol version *015*.

See also

- `msgCOS.ALL`

returnCOS.EXCP

direction: *input*

This message returns *change of state* bits for scheduler exceptions.

To read state of the changed scheduler exception `getSTAT.EXCP` request need to be used.

Remarks

- The message is available since protocol version *015*.

See also

- `getCOS.EXCP`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.FOB

direction: *output*

This message is used to get the COS flags for fob.

The response is `returnCOS.FOB`.

Remarks

- The message is available since protocol version *018*.

See also

- `msgCOS.ALL`

returnCOS.FOB

direction: *input*

This message returns *change of state* bits for fobs.

To read state of the changed fob `getSTAT.FOB` request need to be used.

Remarks

- The message is available since protocol version *018*.

See also

- `getCOS.FOB`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getCOS.CAMERA

direction: *output*

This message is used to get the COS flags for camera.

The response is `returnCOS.CAMERA`.

See also

- `msgCOS.ALL`

returnCOS.CAMERA

direction: *input*

This message returns *change of state* bits for cameras.

To read state of the changed camera `getSTAT.CAMERA` request need to be used.

See also
- `getCOS.CAMERA`

bitSet

multiplicity: *single (static)*
type: *string*

Set of bits, denoting *change of state* for particular objects.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```
the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```
the index of the bit in the byte above

The extracted value of the bit informs whether the status of the particular object has changed (value equal to 1) or not (value equal to 0).

getSTAT.ZONE

direction: *output*

This message is used to get the status bits for the specified zone.

The response is returnSTAT.ZONE.

Remarks

- The message is available since protocol version *011*.

See also

- msgCOS.ALL
- returnCOS.ZONE

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.ZONE

direction: *input*

This message returns ZONE status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.ZONE

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

ZNEV_ACTIVE

multiplicity: *single (static)*
type: *boolean*

Zone in active state.

ZNEV_TAMPER

multiplicity: *single (static)*
type: *boolean*

Zone in tamper state.

ZNEV_AM

multiplicity: *single (static)*
type: *boolean*

Zone in Anti-Mask state.

ZNEV_BATTFAIL

multiplicity: *single (static)*
type: *boolean*

Zone battery fault.

ZNEV_FAULT

multiplicity: *single (static)*
type: *boolean*

Zone in fault state.

ZNEV_DIRTY

multiplicity: *single (static)*
type: *boolean*

Zone in dirty state.

ZNEV_SVSHORT

multiplicity: *single (static)*
type: *boolean*

Zone in supervision short.

ZNEV_SVLONG

multiplicity: *single (static)*
type: *boolean*

Zone in supervision long.

ZNEV_INHIBIT

multiplicity: *single (static)*
type: *boolean*

Zone inhibited.

ZNEV_ISOLATE

multiplicity: *single (static)*
type: *boolean*

Zone isolated.

ZNEV_SOAK

multiplicity: *single (static)*
type: *boolean*

Zone in soak test.

ZNEV_SET

multiplicity: *single (static)*
type: *boolean*

Zone in set mode.

ZNEV_ALARM

multiplicity: *single (static)*
type: *boolean*

Zone in alarm.

ZNEV_LEARNED

multiplicity: *single (static)*
type: *boolean*

Zone has been learned.

Remarks

- The property is available since protocol version *018*.

ZNEV_PRELEARNED

multiplicity: *single (static)*
type: *boolean*

RF signal accept.

Remarks

- The property is available since protocol version *020*.

ZNEV_HELDOPEN

multiplicity: *single (static)*
type: *boolean*

Zone held open too long

Remarks

- The property is available since protocol version *023*.

ZNEV_INVWT

multiplicity: *single (static)*
type: *boolean*

Inverted Walk Test - zone not active for a certain time

Remarks

- The property is available since protocol version *023*.

getSTAT.AREA

direction: *output*

This message is used to get the status bits for the specified area.

The response is returnSTAT.AREA.

Remarks
- The message is available since protocol version *011*.

See also
- msgCOS.ALL
- returnCOS.AREA

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.AREA

direction: *input*

This message returns AREA status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.AREA

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

AREV_FULLSET

multiplicity: *single (static)*
type: *boolean*

Area full set.

AREV_PARTSET

multiplicity: *single (static)*
type: *boolean*

Area part set.

AREV_UNSET

multiplicity: *single (static)*
type: *boolean*

Area unset.

AREV_ALARM

multiplicity: *single (static)*
type: *boolean*

Alarm.

AREV_FSALARM

multiplicity: *single (static)*
type: *boolean*

Area full set and in Alarm.

AREV_PSALARM

multiplicity: *single (static)*
type: *boolean*

Area part set and in Alarm.

AREV_USALARM

multiplicity: *single (static)*
type: *boolean*

Area uset and in Alarm.

AREV_FTCALARM

multiplicity: *single (static)*
type: *boolean*

Area in Alarm and FTC.

AREV_FIREDOOR

multiplicity: *single (static)*
type: *boolean*

Firedoor alarm.

AREV_FSFIREDOOR

multiplicity: *single (static)*
type: *boolean*

Area full set and Firedoor.

AREV_PSFIREDOOR

multiplicity: *single (static)*
type: *boolean*

Area part set and Firedoor.

AREV_USFIREDOOR

multiplicity: *single (static)*
type: *boolean*

Area unset and Firedoor.

AREV_FTCFIREDOOR

multiplicity: *single (static)*
type: *boolean*

Firedoor and FTC.

AREV_FIRE

multiplicity: *single (static)*
type: *boolean*

Fire alarm.

AREV_FSFIRE

multiplicity: *single (static)*
type: *boolean*

Area full set and Fire.

AREV_PSFIRE

multiplicity: *single (static)*
type: *boolean*

Area part set and Fire.

AREV_USFIRE

multiplicity: *single (static)*
type: *boolean*

Area unset and Fire.

AREV_FTCFIRE

multiplicity: *single (static)*
type: *boolean*

Fire and FTC.

AREV_PANIC

multiplicity: *single (static)*
type: *boolean*

Panic alarm.

AREV_FSPANIC

multiplicity: *single (static)*
type: *boolean*

Area full set and Panic.

AREV_PSPANIC

multiplicity: *single (static)*
type: *boolean*

Area part set and Panic.

AREV_USPANIC

multiplicity: *single (static)*
type: *boolean*

Area unset and Panic.

AREV_FTCPANIC

multiplicity: *single (static)*
type: *boolean*

Panic and FTC.

AREV_MEDICAL

multiplicity: *single (static)*
type: *boolean*

Medical alarm.

AREV_FSMEDICAL

multiplicity: *single (static)*
type: *boolean*

Area full set and Medical.

AREV_PSMEDICAL

multiplicity: *single (static)*
type: *boolean*

Area part set and Medical.

AREV_USMEDICAL

multiplicity: *single (static)*
type: *boolean*

Area unset and Medical.

AREV_FTCMEDICAL

multiplicity: *single (static)*
type: *boolean*

Medical and FTC.

AREV_TECHNICAL

multiplicity: *single (static)*
type: *boolean*

Technical alarm.

AREV_FSTECHNICAL

multiplicity: *single (static)*
type: *boolean*

Area full set and Technical.

AREV_PSTECHNICAL

multiplicity: *single (static)*
type: *boolean*

Area part set and Technical.

AREV_USTECHNICAL

multiplicity: *single (static)*
type: *boolean*

Area unset and Technical.

AREV_FTCTECHNICAL

multiplicity: *single (static)*
type: *boolean*

Technical and FTC.

AREV_TAMPER

multiplicity: *single (static)*
type: *boolean*

Tamper alarm.

AREV_FSTAMPER

multiplicity: *single (static)*
type: *boolean*

Area full set and Tamper.

AREV_PSTAMPER

multiplicity: *single (static)*
type: *boolean*

Area part set and Tamper.

AREV_USTAMPER

multiplicity: *single (static)*
type: *boolean*

Area unset and Tamper.

AREV_FTCTAMPER

multiplicity: *single (static)*
type: *boolean*

Tamper and FTC.

AREV_DOORBELL

multiplicity: *single (static)*
type: *boolean*

At least one zone with Doorbell option is active (2sec).

AREV_PSDOORBELL

multiplicity: *single (static)*
type: *boolean*

Area part set and Doorbell.

AREV_USDOORBELL

multiplicity: *single (static)*
type: *boolean*

Area unset and Doorbell.

AREV_ZNACTIVE

multiplicity: *single (static)*
type: *boolean*

At least one zone in Active state.

AREV_ZNINHIBIT

multiplicity: *single (static)*
type: *boolean*

At least one zone in Inhibit state.

AREV_ZNISOLATE

multiplicity: *single (static)*
type: *boolean*

At least one zone in Isolate state.

AREV_ZNFAULT

multiplicity: *single (static)*
type: *boolean*

At least one zone in Fault state.

AREV_ZNAM

multiplicity: *single (static)*
type: *boolean*

At least one zone in AM state.

AREV_ZNTAMPER

multiplicity: *single (static)*
type: *boolean*

At least one zone in Tamper state.

AREV_RASTAMPER

multiplicity: *single (static)*
type: *boolean*

Any RAS in Tamper state.

AREV_RASFAULT

multiplicity: *single (static)*
type: *boolean*

Any RAS in Fault state.

AREV_DGPTAMPER

multiplicity: *single (static)*
type: *boolean*

Any DGP in Tamper state.

AREV_DGPFAULT

multiplicity: *single (static)*
type: *boolean*

Any DGP in Fault state.

AREV_DURESS

multiplicity: *single (static)*
type: *boolean*

Duress code used in the area.

AREV_FSDURESS

multiplicity: *single (static)*
type: *boolean*

Area full set and Duress.

AREV_PSDURESS

multiplicity: *single (static)*
type: *boolean*

Area part set and Duress.

AREV_USDURESS

multiplicity: *single (static)*
type: *boolean*

Area unset and Duress.

AREV_FTCDURESS

multiplicity: *single (static)*
type: *boolean*

Duress and FTC.

AREV_CODETAMPER

multiplicity: *single (static)*
type: *boolean*

Code Tamper from RAS (5sec).

AREV_ENTRY

multiplicity: *single (static)*
type: *boolean*

Area in Entry state.

AREV_EXIT

multiplicity: *single (static)*
type: *boolean*

Area in Exit state.

AREV_EXITFAULT

multiplicity: *single (static)*
type: *boolean*

Alarm detected in Exit state.

AREV_RTS

multiplicity: *single (static)*
type: *boolean*

Area Redy To Set (all conditions set).

AREV_SETOK

multiplicity: *single (static)*
type: *boolean*

Exit time finished without alarms (30sec).

AREV_SETFAULT

multiplicity: *single (static)*
type: *boolean*

Set not possible (30sec).

AREV_UNSETOK

multiplicity: *single (static)*
type: *boolean*

Area unset OK (30sec).

AREV_ALARMACK

multiplicity: *single (static)*
type: *boolean*

Alarms are waiting for ACK.

AREV_FIRERESET

multiplicity: *single (static)*
type: *boolean*

Active after Fire alarm ACK (10sec).

AREV_WALK

multiplicity: *single (static)*
type: *boolean*

Area in Walk-Test.

AREV_WALKZNACTV

multiplicity: *single (static)*
type: *boolean*

Zone in Walk-Test is activated.

AREV_AALARM

multiplicity: *single (static)*
type: *boolean*

A alarm from ACPO (reset after unset).

AREV_BALARM

multiplicity: *single (static)*
type: *boolean*

B alarm from ACPO (reset after unset).

AREV_ISIREN

multiplicity: *single (static)*
type: *boolean*

Internal Siren active (retriggering allowed).

AREV_ESIREN

multiplicity: *single (static)*
type: *boolean*

External Siren active (no retriggering).

AREV_STROBE

multiplicity: *single (static)*
type: *boolean*

Strobe Output active (clear after unset).

AREV_BUZZER

multiplicity: *single (static)*
type: *boolean*

Buzzer Output active.

AREV_AMRESET

multiplicity: *single (static)*
type: *boolean*

Anti mask triggers Walktest on set attempt.

Remarks
- The property is available since protocol version *013*.

AREV_PARTSET2

multiplicity: *single (static)*
type: *boolean*

Area part set 2.

Remarks
- The property is available since protocol version *015*.

AREV_WARNING

multiplicity: *single (static)*
type: *boolean*

Area warning.

Remarks
- The property is available since protocol version *015*.

AREV_AUTOARM

multiplicity: *single (static)*
type: *boolean*

Area autoarm.

Remarks
- The property is available since protocol version *015*.

AREV_HAALARM

multiplicity: *single (static)*
type: *boolean*

HA alarm from ACPO (reset after unset).

AREV_HBALARM

multiplicity: *single (static)*
type: *boolean*

HB alarm from ACPO (reset after unset).

getSTAT.RAS

direction: *output*

This message is used to get the status bits for the specified keypad.

The response is returnSTAT.RAS.

Remarks
- The message is available since protocol version *011*.

See also
- msgCOS.ALL
- returnCOS.RAS

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.RAS

direction: *input*

This message returns keypad status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.RAS

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

RASEV_OFFLINE

multiplicity: *single (static)*
type: *boolean*

RAS is offline.

RASEV_RTE

multiplicity: *single (static)*
type: *boolean*

RAS RTE input is active.

RASEV_CODETAMPER

multiplicity: *single (static)*
type: *boolean*

Code tamper.

RASEV_TAMPER

multiplicity: *single (static)*
type: *boolean*

Tamper active.

RASEV_DURESS

multiplicity: *single (static)*
type: *boolean*

Duress code entered.

RASEV_CARD

multiplicity: *single (static)*
type: *boolean*

Active Card (3sec).

RASEV_PIN

multiplicity: *single (static)*
type: *boolean*

Valid PIN (also Duress) (3sec).

RASEV_DOORACC

multiplicity: *single (static)*
type: *boolean*

Door access (3sec).

RASEV_LOCKED

multiplicity: *single (static)*
type: *boolean*

RAS Locked for 120sec (code tamper or set by keyswitch).

RASEV_ISOLATE

multiplicity: *single (static)*
type: *boolean*

RAS Isolated.

RASEV_DOORBELL

multiplicity: *single (static)*
type: *boolean*

RAS Doorbell state.

Remarks

- The property is available since protocol version *019*.

RASEV_CARDV

multiplicity: *single (static)*
type: *boolean*

Valid Card (3sec).

Remarks

- The property is available since protocol version *019*.

RASEV_EXIT_START

multiplicity: *single (static)*
type: *boolean*

Exit time started on RAS (3sec).

RASEV_ENTRY_STOPPED

multiplicity: *single (static)*
type: *boolean*

Entry time stopped on RAS (3sec).

getSTAT.DGP

direction: *output*

This message is used to get the status bits for the specified expander.

The response is `returnSTAT.DGP`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`
- `returnCOS.DGP`

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.DGP

direction: *input*

This message returns expander status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.DGP

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

DGPEV_OFFLINE

multiplicity: *single (static)*
type: *boolean*

DGP is offline.

DGPEV_MAINSFAIL

multiplicity: *single (static)*
type: *boolean*

DGP mains fail.

DGPEV_BATTFAIL

multiplicity: *single (static)*
type: *boolean*

DGP battery fail.

DGPEV_TAMPER

multiplicity: *single (static)*
type: *boolean*

DGP Tamper active.

DGPEV_FUSEFAULT

multiplicity: *single (static)*
type: *boolean*

DGP Fuse fault.

DGPEV_SIRENFAULT

multiplicity: *single (static)*
type: *boolean*

DGP Siren fault.

DGPEV_RCVFAULT

multiplicity: *single (static)*
type: *boolean*

DGP Receiver fault.

DGPEV_ISOLATE

multiplicity: *single (static)*
type: *boolean*

DGP Isolated.

DGPEV_BATTLOW

multiplicity: *single (static)*
type: *boolean*

DGP Battery is Low or Missing.

DGPEV_BTESTACTV

multiplicity: *single (static)*
type: *boolean*

DGP battery test active.

DGPEV_BTESTFAIL

multiplicity: *single (static)*
type: *boolean*

DGP battery test fail.

DGPEV_PU_FAIL

multiplicity: *single (static)*
type: *boolean*

DGP Power unit fail.

getSTAT.DGP0

direction: *output*

This message is used to get the status bits for DGP0.

The response is `returnSTAT.DGP0`.

Remarks

- The message is available since protocol version *011*.
- The *index* property must be set to 0.

See also

- `msgCOS.ALL`

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.DGP0

direction: *input*

This message returns DGP0 status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.DGP0

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

DGP0EV_MAINSFAIL

multiplicity: *single (static)*
type: *boolean*

DGP0 Mains fail.

DGP0EV_BATTFAIL

multiplicity: *single (static)*
type: *boolean*

DGP0 Battery fail.

DGP0EV_TAMPER

multiplicity: *single (static)*
type: *boolean*

DGP0 tamper.

DGP0EV_FUSEFAULT

multiplicity: *single (static)*
type: *boolean*

DGP0 Fuse fault.

DGP0EV_SIRENFAULT

multiplicity: *single (static)*
type: *boolean*

DGP0 Siren fault.

DGP0EV_LF

multiplicity: *single (static)*
type: *boolean*

DGP0 Line Fault.

DGP0EV_LFPSTN

multiplicity: *single (static)*
type: *boolean*

DGP0 Line Fault (PSTN).

DGP0EV_LFISDN

multiplicity: *single (static)*
type: *boolean*

DGP0 Line Fault (ISDN).

DGP0EV_LFGSM

multiplicity: *single (static)*
type: *boolean*

DGP0 Line Fault (GSM).

DGP0EV_FTC

multiplicity: *single (static)*
type: *boolean*

DGP0 FTC.

DGP0EV_MIFAULT

multiplicity: *single (static)*
type: *boolean*

DGP0 MI device fault.

DGP0EV_MIFISDN

multiplicity: *single (static)*
type: *boolean*

DGP0 MI device fault (ISDN).

DGP0EV_MIFGSM

multiplicity: *single (static)*
type: *boolean*

DGP0 MI device fault (GSM).

DGP0EV_MIFVOICE

multiplicity: *single (static)*
type: *boolean*

DGP0 MI device fault (Voice).

DGP0EV_NTPF

multiplicity: *single (static)*
type: *boolean*

NTP server fault.

DGP0EV_LFETH

multiplicity: *single (static)*
type: *boolean*

DGP0 Line Fault (Ethernet interface).

DGP0EV_LFIP

multiplicity: *single (static)*
type: *boolean*

DGP0 Line Fault (IP configuration).

DGP0EV_LFGPRS

multiplicity: *single (static)*
type: *boolean*

DGP0 Line Fault (GPRS configuration).

DGP0EV_LFIPGPRS

multiplicity: *single (static)*
type: *boolean*

DGP0 Line Fault (GPRS configuration).

DGP0EV_LFTDA

multiplicity: *single (static)*
type: *boolean*

DGP0 Line Fault (TDA module).

Remarks
- The property is available since protocol version *020*.

DGP0EV_LFTDAGPRS

multiplicity: *single (static)*
type: *boolean*

DGP0 Line Fault (TDA module - GPRS).

Remarks
- The property is available since protocol version *020*.

DGP0EV_LFTDAETH

multiplicity: *single (static)*
type: *boolean*

DGP0 Line Fault (TDA module - Ethernet interface).

Remarks
- The property is available since protocol version *020*.

DGP0EV_MIFTDA

multiplicity: *single (static)*
type: *boolean*

DGP0 MI device fault (TDA module).

Remarks
- The property is available since protocol version *020*.

getSTAT.OUT

direction: *output*

This message is used to get the status bits for the specified OUTPUT.

The response is returnSTAT.OUT.

Remarks

- The message is available since protocol version *011*.

See also

- msgCOS.ALL
- returnCOS.OUT

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.OUT

direction: *input*

This message returns OUTPUT status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.OUT

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

OUTEV_ACTIVE

multiplicity: *single (static)*
type: *boolean*

Output state is ACTIVE.

OUTEV_ON

multiplicity: *single (static)*
type: *boolean*

Output state is ON.

getSTAT.FILTER

direction: *output*

This message is used to get the status bits for the specified FILTER.

The response is returnSTAT.FILTER.

Remarks

- The message is available since protocol version *011*.

See also

- msgCOS.ALL
- returnCOS.FILTER

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.FILTER

direction: *input*

This message returns FILTER status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.FILTER

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

CFLEV_ACTIVE

multiplicity: *single (static)*
type: *boolean*

Filter output state is ACTIVE.

getSTAT.PCC

direction: *output*

This message is used to get the status bits for the specified PC connection.

The response is returnSTAT.PCC.

Remarks

- The message is available since protocol version *011*.

See also

- msgCOS.ALL
- returnCOS.PCC

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.PCC

direction: *input*

This message returns PC connection status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.PCC

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

PCCEV_RUNNING

multiplicity: *single (static)*
type: *boolean*

PCC is in active state (running).

getSTAT.SYS

direction: *output*

This message is used to get the status bits for SYSTEM object.

The response is returnSTAT.SYS.

Remarks

- The message is available since protocol version *011*.

See also

- msgCOS.ALL

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.SYS

direction: *input*

This message returns SYSTEM status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.SYS

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

SYSEV_ALLSET

multiplicity: *single (static)*
type: *boolean*

All Areas are Set.

SYSEV_AUTOANS

multiplicity: *single (static)*
type: *boolean*

Auto Answer procedure started (1min).

SYSEV_RCONNACTV

multiplicity: *single (static)*
type: *boolean*

Remote Computer connection active.

SYSEV_RCONNFAIL

multiplicity: *single (static)*
type: *boolean*

Remote Computer connection failed (1min).

SYSEV_LPRGACTV

multiplicity: *single (static)*
type: *boolean*

Local programming active (installer present).

SYSEV_RPRGACTV

multiplicity: *single (static)*
type: *boolean*

Remote programming active (installer present).

SYSEV_TIMECHG

multiplicity: *single (static)*
type: *boolean*

System Time changed (3sec).

SYSEV_SSAVER

multiplicity: *single (static)*
type: *boolean*

Screen Saver active (system wide).

SYSEV_ISIREN

multiplicity: *single (static)*
type: *boolean*

Internal Siren active (retriggering allowed).

SYSEV_ESIREN

multiplicity: *single (static)*
type: *boolean*

External Siren active (no retriggering).

SYSEV_STROBE

multiplicity: *single (static)*
type: *boolean*

Strobe Output active clear after unset.

SYSEV_FAULT

multiplicity: *single (static)*
type: *boolean*

Global system Fault flag.

SYSEV_TAMPER

multiplicity: *single (static)*
type: *boolean*

Global system Tamper flag.

SYSEV_SERVICEIN

multiplicity: *single (static)*
type: *boolean*

Service In active.

Remarks

- The property is available since protocol version *013*.

SYSEV_WALKTST_MODE

multiplicity: *single (static)*
type: *boolean*

Wakltest mode

Remarks

- The property is available since protocol version *023*.

getSTAT.CS

direction: *output*

This message is used to get the status bits for the specified CS.

The response is `returnSTAT.CS`.

Remarks

- The message is available since protocol version *011*.

See also

- `msgCOS.ALL`
- `returnCOS.CS`

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.CS

direction: *input*

This message returns CS status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.CS

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

CSEV_FTC

multiplicity: *single (static)*
type: *boolean*

FTC state active.

CSEV_HBF

multiplicity: *single (static)*
type: *boolean*

Heartbeat fault (IP only).

CSEV_BUSY

multiplicity: *single (static)*
type: *boolean*

CS is in use.

getSTAT.TRIGG

direction: *output*

This message is used to get the status bits for the specified TRIGGER.

The response is returnSTAT.TRIGG.

Remarks

- The message is available since protocol version *011*.

See also

- msgCOS.ALL
- returnCOS.TRIGG

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.TRIGG

direction: *input*

This message returns TRIGGER status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.TRIGG

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

TRGEV_KEYFOBSW1

multiplicity: *single (static)*
type: *boolean*

Key Fob switch 1 toggle.

TRGEV_KEYFOBSW2

multiplicity: *single (static)*
type: *boolean*

Key Fob switch 2 toggle.

TRGEV_KEYFOBSW12

multiplicity: *single (static)*
type: *boolean*

Key Fob switch 1 press changes to ON, Key Fob switch 2 changes to OFF.

TRGEV_REMOTEOUT

multiplicity: *single (static)*
type: *boolean*

Remote Output control OFF/ON.

TRGEV_FKEY

multiplicity: *single (static)*
type: *boolean*

Function key control.

Remarks
- The property is available since protocol version *015*.

TRGEV_SCHEDULE

multiplicity: *single (static)*
type: *boolean*

Schedule control.

Remarks
- The property is available since protocol version *015*.

TRGEV_FOB

multiplicity: *single (static)*
type: *boolean*

Fob control.

Remarks
- The property is available since protocol version *018*.

getSTAT.USER

direction: *output*

This message is used to get the status bits for the specified USER.

The response is returnSTAT.USER.

Remarks
- The message is available since protocol version *011*.

See also
- msgCOS.ALL
- returnCOS.USER

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.USER

direction: *input*

This message returns USER status bits.

Remarks

- The message is available since protocol version *011*.

See also

- getSTAT.USER

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

USEREV_CARDPIN

multiplicity: *single (static)*
type: *boolean*

Valid Card/PIN presented by the user.

USEREV_SMSCTRLACTIVE

multiplicity: *single (static)*
type: *boolean*

SMS control active.

USEREV_SMSCTRLLOCK

multiplicity: *single (static)*
type: *boolean*

SMS control disabled by invalid attempts.

USEREV_SMSREPACTIVE

multiplicity: *single (static)*
type: *boolean*

SMS reporting active.

USEREV_SMSREPAFTERSET

multiplicity: *single (static)*
type: *boolean*

SMS reporting will be activated after set.

getSTAT.UG

direction: *output*

This message is used to get the status bits for the specified USER GROUP.

The response is returnSTAT.UG (status bits)

Remarks

- The message is available since protocol version *011*.

See also

- msgCOS.ALL
- returnCOS.UG

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.UG

direction: *input*

This message returns USER GROUP status bits.

Remarks

- The message is available since protocol version *013*.

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

UGEV_CARDPIN

multiplicity: *single (static)*
type: *boolean*

Valid Card/PIN presented by the user.

getSTAT.EXCP

direction: *output*

This message is used to get the status bits for the specified schedule exception.

The response is returnSTAT.EXCP.

Remarks

- The message is available since protocol version *015*.

See also

- msgCOS.ALL
- returnCOS.EXCP

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.EXCP

direction: *input*

This message returns schedule exception status bits.

Remarks

- The message is available since protocol version *015*.

See also

- getSTAT.EXCP

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

EXCPEV_ACTIVE

multiplicity: *single (static)*
type: *boolean*

Exception is activated.

getSTAT.SCAL

direction: *output*

This message is used to get the status bits for the specified schedule calendar.

The response is returnSTAT.SCAL.

Remarks

- The message is available since protocol version *015*.

See also

- msgCOS.ALL

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.SCAL

direction: *input*

This message returns schedule calendar status bits.

Remarks

- The message is available since protocol version *015*.

See also

- getSTAT.SCAL

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

SCALEV_HOUR

multiplicity: *single (static)*
type: *boolean*

Toggled every hour.

SCALEV_DAY

multiplicity: *single (static)*
type: *boolean*

Toggled every day.

SCALEV_MON

multiplicity: *single (static)*
type: *boolean*

Active on Monday.

SCALEV_TUE

multiplicity: *single (static)*
type: *boolean*

Active on Tuesday.

SCALEV_WED

multiplicity: *single (static)*
type: *boolean*

Active on Wednesday.

SCALEV_THU

multiplicity: *single (static)*
type: *boolean*

Active on Thursday.

SCALEV_FRI

multiplicity: *single (static)*
type: *boolean*

Active on Friday.

SCALEV_SAT

multiplicity: *single (static)*
type: *boolean*

Active on Saturday.

SCALEV_SUN

multiplicity: *single (static)*
type: *boolean*

Active on Sunday.

getSTAT.FOB

direction: *output*

This message is used to get the status bits for the specified fob.

The response is returnSTAT.FOB.

Remarks
- The message is available since protocol version *018*.

See also
- msgCOS.ALL
- returnCOS.FOB

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.FOB

direction: *input*

This message returns fob status bits.

Remarks

- The message is available since protocol version *018*.

See also

- getSTAT.FOB

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

FOBEV_LEARNED

multiplicity: *single (static)*
type: *boolean*

Fob has been learned.

FOBEV_BUTTON1

multiplicity: *single (static)*
type: *boolean*

Fob Button 1 has been pressed.

FOBEV_BUTTON2

multiplicity: *single (static)*
type: *boolean*

Fob Button 2 has been pressed.

FOBEV_BUTTON3

multiplicity: *single (static)*
type: *boolean*

Fob Button 3 has been pressed.

FOBEV_BUTTON4

multiplicity: *single (static)*
type: *boolean*

Fob Button 4 has been pressed.

FOBEV_BUTTON12

multiplicity: *single (static)*
type: *boolean*

Fob Button 12 has been pressed.

FOBEV_BUTTON13

multiplicity: *single (static)*
type: *boolean*

Fob Button 13 has been pressed.

FOBEV_BUTTON14

multiplicity: *single (static)*
type: *boolean*

Fob Button 14 has been pressed.

FOBEV_BUTTON23

multiplicity: *single (static)*
type: *boolean*

Fob Button 23 has been pressed.

FOBEV_BUTTON24

multiplicity: *single (static)*
type: *boolean*

Fob Button 24 has been pressed.

FOBEV_BUTTON34

multiplicity: *single (static)*
type: *boolean*

Fob Button 34 has been pressed.

getSTAT.CAMERA

direction: *output*

This message is used to get the status bits for the specified fob.

The response is returnSTAT.CAMERA.

See also

- msgCOS.ALL
- returnCOS.CAMERA

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

returnSTAT.CAMERA

direction: *input*

This message returns fob status bits.

See also

- getSTAT.CAMERA

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

CAMEV_PICTURE_CAPTURED

multiplicity: *single (static)*
type: *boolean*

A picture has been captured.

CAMEV_EV_LIMIT_EXCEEDED

multiplicity: *single (static)*
type: *boolean*

Picture taken limit reached.

createCC.A_STATE

direction: *output*

This message is used to initiate the CC session for area arming preview.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial and only state of the successfully created CC session is `SC_Active`. The status can be read using `statusCC.SESSION` request.

The following actions are available in the only CC session state:

- `SC_Active`

  Accepted requests:
  - `fnCC.A_STATE_GET_INH`
  - `fnCC.A_STATE_GET_UNINH`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and can accept all control methods that list causes that make the areas not ready for arm.

Remarks

- The message is available since protocol version *021*.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

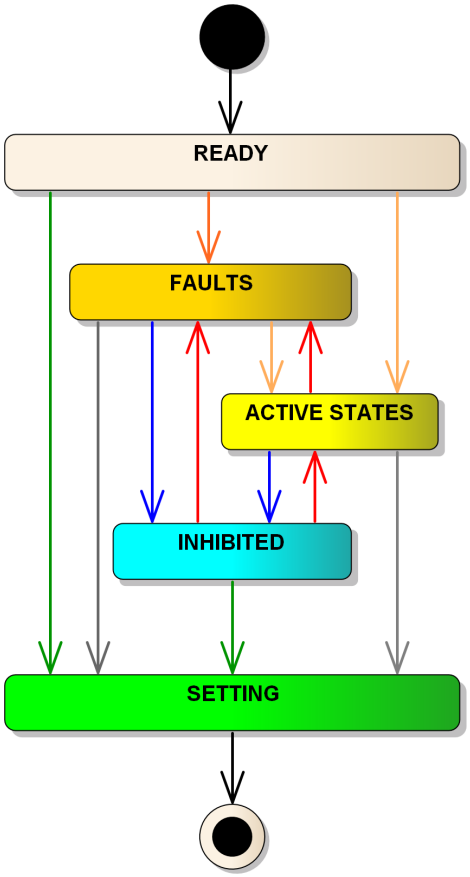
createCC.A_SET

direction: *output*

This message is used to initiate the CC session for area arming.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The state machine of the setting process is shown on the diagram below:



The initial state of the successfully created CC session is `SC_Ready`. The inital status can be read using `statusCC.SESSION` request.

Depending on the current CC session state the following actions are available:

- `SC_Ready`

  Control task is created and ready to start.

The request fnCC.A_SET_SETAREAS is used to proceed with arming the areas for the current CC session. If there is no problems to start setting the state machine goes to SC_Setting state as it is shown with green arrow on the diagram. However, if there are active faults or active zones, then the state machine goes to SC_Faults or SC_ActiveStates accordingly, which is indicated with dark and light orange arrows.

Accepted requests:
- fnCC.A_SET_SETAREAS
- statusCC.SESSION
- destroyCC.SESSION

- SC_Faults

  There are detected faults in the system that prevents setting selected areas.

  If user is allowed to inhibit them there is possible to read the list of the faults using fnCC.A_SET_GETFAULT requests. The expected responses with the fault data are return.sysevent. There is a property eventUniqueID in the received fault that need to be used as a seletor to inhibit the event with fnCC.A_SET_INHFAULT requests. As all faults are inhibited, the state machine goes to SC_Inhibited state (the blue arrow in the diagram). However, if there are active zones, then the state machine goes to SC_ActiveStates (the light orange arrow in the diagram).

  If user is allowed to set system in force mode, fnCC.A_SET_FORCEDSET request can be used (the gray arrow in the diagram).

  Notes:
  - There is possible that the state machine is still in this state even the all and once collected faults are inhibited. In that case, new faults are detected in the system. The list of faults need to be refreshed and the new faults need to be inhibited.

  Accepted requests:
  - fnCC.A_SET_GETFAULT
  - fnCC.A_SET_INHFAULT
  - fnCC.A_SET_FORCEDSET
  - statusCC.SESSION
  - destroyCC.SESSION

- SC_ActiveStates

  There are detected active states in zones or devices that prevents setting selected areas.

If user is allowed to inhibit them there is possible to read the list of the active states using `fnCC.A_SET_GETACTIVE` requests. The expected responses with the active state data are `return.sysevent`. There is a property `eventUniqueID` in the received active state that need to be used as a seletor to inhibit the event with `fnCC.A_SET_INHACTIVE` requests. As all active states are inhibited, the state machine goes to `SC_Inhibited` state (the bluee arrow in the diagram).

If user is allowed to set system in force mode, `fnCC.A_SET_FORCEDSET` request can be used (the gray arrow in the diagram).

Notes:

- There is possible that the state machine is still in this state even the all and once collected active states are inhibited. In that case, new active states are detected in the system. The list of active states need to be refreshed and the new active states need to be inhibited.
- There is possible that the state machine returns back to `SC_Faults` state (the read arrow in the diagram). In that case, new faults are detected in the system.

Accepted requests:

- `fnCC.A_SET_GETACTIVE`
- `fnCC.A_SET_INHACTIVE`
- `fnCC.A_SET_FORCEDSET`
- `statusCC.SESSION`
- `destroyCC.SESSION`

- `SC_Inhibited`

There are inhibited/isolated zones/devices from actions performed in `SC_Faults` or `SC_ActiveStates` states.

In this state the control task is waiting for user confirmation to start the setting process. The list of events that need to be confirmed can be read with `fnCC.A_SET_GETINHIB` requests. The request `fnCC.A_SET_SETAREAS` is used to proceed with arming the areas for the current CC session (the green arrow in the diagram).

Notes:

- There is possible that the state machine returns back to `SC_Faults` or `SC_ActiveStates` state (the read arrows in the diagram). In that case, new faults or active states are detected in the system.

Accepted requests:

- `fnCC.A_SET_GETINHIB`

  - `fnCC.A_SET_SETAREAS`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

- `SC_Setting`

  Check areas before setting and start setting selected process if all is OK.

  Accepted requests:
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

Remarks
- The message is available since protocol version *011*.

`area.1`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

`area.2`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

`area.3`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

`area.4`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

`area.5`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.A_CONFAL

direction: *output*

This message is used to initiate the CC session for alarm confirming.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial state of the successfully created CC session is `CC_Ready`. The inital status can be read using `statusCC.SESSION` request.

Depending on the current CC session state the following actions are available:

- `CC_Ready`

  Accepted requests:
  - `fnCC.A_CONFAL_START`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and ready to start. The request `fnCC.A_CONFAL_START` is used to proceed with confirming alarms for the current CC session.

- `CC_CnfAlarms`

  Accepted requests:
  - `fnCC.A_CONFAL_GETALARM`
  - `fnCC.A_CONFAL_CONFALARM`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Alarms detected that must be confirmed by the user. It is possible to check and confirm alarms (if allowed for the user). If there are no more alarms to confirm, state is changed to `CC_Confirmed`.

- `CC_Confirmed`

  Accepted requests:
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  All alarms confirmed.

Remarks

- The message is available since protocol version *011*.

`area.1`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

`area.2`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

`area.3`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

`area.4`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

`area.5`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

`area.6`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.A_WALKTST

direction: *output*

This message is used to initiate the CC session for walk test.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial state of the successfully created CC session is `WC_Ready`. The inital status can be read using `statusCC.SESSION` request.

Depending on the CC session state the following actions are possible:

- `WC_Ready`

  Accepted requests:
  - `fnCC.A_WALKTST_START`
  - `fnCC.A_WALKTST_GETLIST`
  - `fnCC.A_WALKTST_START_WITH_REP`
  - `fnCC.A_WALKTST_ADD_ZONE`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and ready to start. If walk test is created with non-empty list of areas there is possible to check the list of events that will be walk tested using the `fnCC.A_WALKTST_GETLIST` request. Otherwise the list of events that will be walk testet must be created manually using the `fnCC.A_WALKTST_ADD_ZONE` request.

- `WC_Active`

  Accepted requests:
  - `fnCC.A_WALKTST_GETEV`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Walk test process is active. In this state it is possible to get walk test points already tested. If there are no more points to walk test or time-out error occurred, state is changed to `WC_Finished`.

- `WC_Finished`

  Accepted requests:

- `fnCC.A_WALKTST_GETRES`
- `statusCC.SESSION`
- `destroyCC.SESSION`

Walk test process finished. Walk test process result can be checked by `fnCC.A_WALKTST_GETRES` request to list of not tested points.

Remarks

- The message is available since protocol version *011*.

`area.1`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

`area.2`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

`area.3`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

`area.4`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

`area.5`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.A_UNSET

direction: *output*

This message is used to initiate the CC session for area unsetting.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial state of the successfully created CC session is `UC_Ready`. The inital status can be read using `statusCC.SESSION` request.

Depending on the current CC session state the following actions are available:

- `UC_Ready`

  Accepted requests:
  - `fnCC.A_UNSET_UNSETAREAS`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and ready to start. The request `fnCC.A_UNSET_UNSETAREAS` is used to proceed with unsetting the areas for the current CC session.

- `UC_Unsetting`

  Accepted requests:
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Unset is in progress. After all areas are unset state is changed to `UC_CnfAlarms`.

- `UC_CnfAlarms`

  Accepted requests:
  - `fnCC.A_UNSET_GETALARM`
  - `fnCC.A_UNSET_CONFALARM`
  - `fnCC.A_UNSET_SKIP`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

Alarms detected that must be confirmed by the user. It is possible to check and confirm alarms (if allowed for the user). If there are no more alarms to confirm, state is changed to `UC_CnfFaults`. It is possible to skip this state (leave alarms not confirmed).

- `UC_CnfFaults`

  Accepted requests:
  - `fnCC.A_UNSET_GETFAULT`
  - `fnCC.A_UNSET_CONFFAULT`
  - `fnCC.A_UNSET_SKIP`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Faults detected that should be confirmed by the user. It is possible to check and confirm faults. If there are no more faults to confirm, state is changed to `UC_Unset`. It is possible to skip this state (leave faults not confirmed).

- `UC_Unset`

  Accepted requests:
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  All selected areas are unset.

Remarks
- The message is available since protocol version *011*.

`area.1`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

`area.2`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

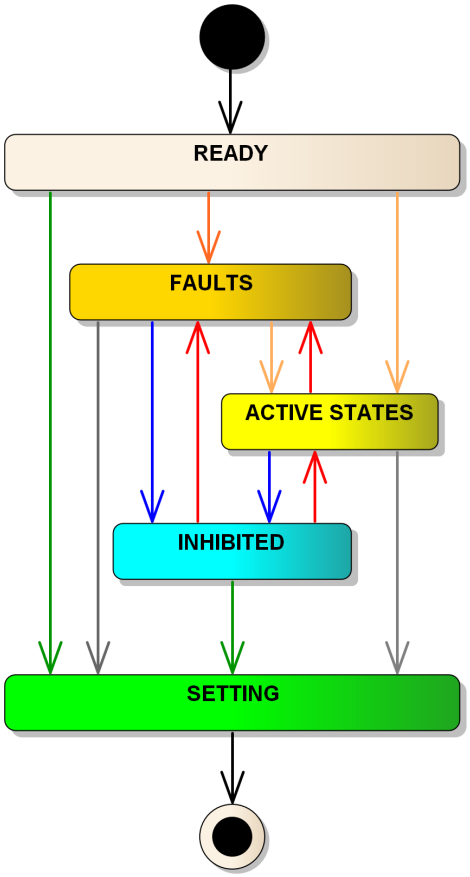
createCC.A_PARTSET

direction: *output*

This message is used to initiate the CC session for area part set.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The state machine of the setting process is shown on the diagram below:



The initial state of the successfully created CC session is `PS_Ready`. The inital status can be read using `statusCC.SESSION` request.

Depending on the current CC session state the following actions are available:

- `PS_Ready`

  Control task is created and ready to start.

The request fnCC.A_SET_SETAREAS is used to proceed with arming the areas for the current CC session. If there is no problems to start setting the state machine goes to PS_Setting state as it is shown with green arrow on the diagram. However, if there are active faults or active zones, then the state machine goes to PS_Faults or PS_ActiveStates accordingly, which is indicated with dark and light orange arrows.

Accepted requests:
- fnCC.A_SET_SETAREAS
- statusCC.SESSION
- destroyCC.SESSION

- PS_Faults

  There are detected faults in the system that prevents setting selected areas.

  If user is allowed to inhibit them there is possible to read the list of the faults using fnCC.A_SET_GETFAULT requests. The expected responses with the fault data are return.sysevent. There is a property eventUniqueID in the received fault that need to be used as a seletor to inhibit the event with fnCC.A_SET_INHFAULT requests. As all faults are inhibited, the state machine goes to PS_Inhibited state (the blue arrow in the diagram). However, if there are active zones, then the state machine goes to PS_ActiveStates (the light orange arrow in the diagram).

  If user is allowed to set system in force mode, fnCC.A_SET_FORCEDSET request can be used (the gray arrow in the diagram).

  Notes:
  - There is possible that the state machine is still in this state even the all and once collected faults are inhibited. In that case, new faults are detected in the system. The list of faults need to be refreshed and the new faults need to be inhibited.

  Accepted requests:
  - fnCC.A_SET_GETFAULT
  - fnCC.A_SET_INHFAULT
  - fnCC.A_SET_FORCEDSET
  - statusCC.SESSION
  - destroyCC.SESSION

- PS_ActiveStates

  There are detected active states in zones or devices that prevents setting selected areas.

If user is allowed to inhibit them there is possible to read the list of the active states using `fnCC.A_SET_GETACTIVE` requests. The expected responses with the active state data are `return.sysevent`. There is a property `eventUniqueID` in the received active state that need to be used as a seletor to inhibit the event with `fnCC.A_SET_INHACTIVE` requests. As all active states are inhibited, the state machine goes to `PS_Inhibited` state (the bluee arrow in the diagram).

If user is allowed to set system in force mode, `fnCC.A_SET_FORCEDSET` request can be used (the gray arrow in the diagram).

Notes:

- There is possible that the state machine is still in this state even the all and once collected active states are inhibited. In that case, new active states are detected in the system. The list of active states need to be refreshed and the new active states need to be inhibited.
- There is possible that the state machine returns back to `PS_Faults` state (the read arrow in the diagram). In that case, new faults are detected in the system.

Accepted requests:

- `fnCC.A_SET_GETACTIVE`
- `fnCC.A_SET_INHACTIVE`
- `fnCC.A_SET_FORCEDSET`
- `statusCC.SESSION`
- `destroyCC.SESSION`

- `PS_Inhibited`

There are inhibited/isolated zones/devices from actions performed in `PS_Faults` or `PS_ActiveStates` states.

In this state the control task is waiting for user confirmation to start the setting process. The list of events that need to be confirmed can be read with `fnCC.A_SET_GETINHIB` requests. The request `fnCC.A_SET_SETAREAS` is used to proceed with arming the areas for the current CC session (the green arrow in the diagram).

Notes:

- There is possible that the state machine returns back to `PS_Faults` or `PS_ActiveStates` state (the read arrows in the diagram). In that case, new faults or active states are detected in the system.

Accepted requests:

- `fnCC.A_SET_GETINHIB`

  - `fnCC.A_SET_SETAREAS`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

- `PS_Setting`

  Check areas before setting and start setting selected process if all is OK.

  Accepted requests:
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

Remarks
- The message is available since protocol version *011*.

`area.1`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

`area.2`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

`area.3`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

`area.4`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

`area.5`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

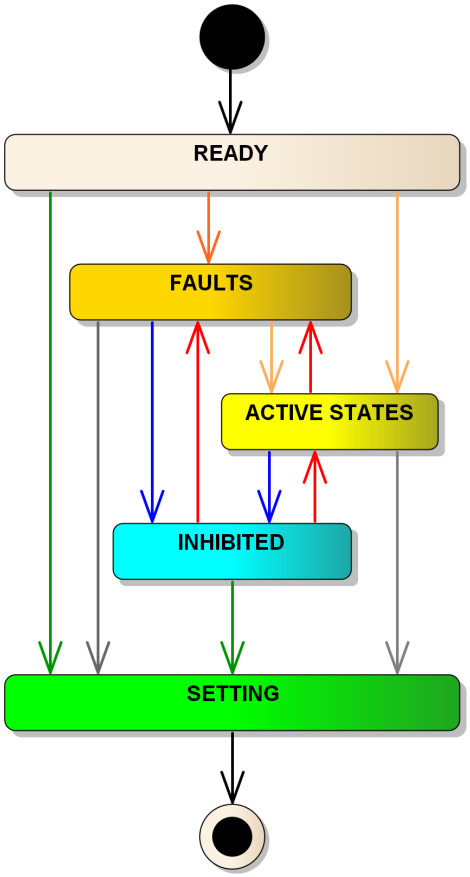
createCC.A_PARTSET2

direction: *output*

This message is used to initiate the CC session for area part set 2.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The state machine of the setting process is shown on the diagram below:



The initial state of the successfully created CC session is `PS2_Ready`. The inital status can be read using `statusCC.SESSION` request.

Depending on the current CC session state the following actions are available:

- `PS2_Ready`

  Control task is created and ready to start.

The request fnCC.A_SET_SETAREAS is used to proceed with arming the areas for the current CC session. If there is no problems to start setting the state machine goes to PS2_Setting state as it is shown with green arrow on the diagram. However, if there are active faults or active zones, then the state machine goes to PS2_Faults or PS2_ActiveStates accordingly, which is indicated with dark and light orange arrows.

Accepted requests:
- fnCC.A_SET_SETAREAS
- statusCC.SESSION
- destroyCC.SESSION

- PS2_Faults

  There are detected faults in the system that prevents setting selected areas.

  If user is allowed to inhibit them there is possible to read the list of the faults using fnCC.A_SET_GETFAULT requests. The expected responses with the fault data are return.sysevent. There is a property eventUniqueID in the received fault that need to be used as a seletor to inhibit the event with fnCC.A_SET_INHFAULT requests. As all faults are inhibited, the state machine goes to PS2_Inhibited state (the blue arrow in the diagram). However, if there are active zones, then the state machine goes to PS2_ActiveStates (the light orange arrow in the diagram).

  If user is allowed to set system in force mode, fnCC.A_SET_FORCEDSET request can be used (the gray arrow in the diagram).

  Notes:
  - There is possible that the state machine is still in this state even the all and once collected faults are inhibited. In that case, new faults are detected in the system. The list of faults need to be refreshed and the new faults need to be inhibited.

  Accepted requests:
  - fnCC.A_SET_GETFAULT
  - fnCC.A_SET_INHFAULT
  - fnCC.A_SET_FORCEDSET
  - statusCC.SESSION
  - destroyCC.SESSION

- PS2_ActiveStates

  There are detected active states in zones or devices that prevents setting selected areas.

If user is allowed to inhibit them there is possible to read the list of the active states using `fnCC.A_SET_GETACTIVE` requests. The expected responses with the active state data are `return.sysevent`. There is a property `eventUniqueID` in the received active state that need to be used as a seletor to inhibit the event with `fnCC.A_SET_INHACTIVE` requests. As all active states are inhibited, the state machine goes to `PS2_Inhibited` state (the bluee arrow in the diagram).

If user is allowed to set system in force mode, `fnCC.A_SET_FORCEDSET` request can be used (the gray arrow in the diagram).

Notes:

- There is possible that the state machine is still in this state even the all and once collected active states are inhibited. In that case, new active states are detected in the system. The list of active states need to be refreshed and the new active states need to be inhibited.
- There is possible that the state machine returns back to `PS2_Faults` state (the read arrow in the diagram). In that case, new faults are detected in the system.

Accepted requests:

- `fnCC.A_SET_GETACTIVE`
- `fnCC.A_SET_INHACTIVE`
- `fnCC.A_SET_FORCEDSET`
- `statusCC.SESSION`
- `destroyCC.SESSION`

- `PS2_Inhibited`

There are inhibited/isolated zones/devices from actions performed in `PS2_Faults` or `PS2_ActiveStates` states.

In this state the control task is waiting for user confirmation to start the setting process. The list of events that need to be confirmed can be read with `fnCC.A_SET_GETINHIB` requests. The request `fnCC.A_SET_SETAREAS` is used to proceed with arming the areas for the current CC session (the green arrow in the diagram).

Notes:

- There is possible that the state machine returns back to `PS2_Faults` or `PS2_ActiveStates` state (the read arrows in the diagram). In that case, new faults or active states are detected in the system.

Accepted requests:

- `fnCC.A_SET_GETINHIB`

  - `fnCC.A_SET_SETAREAS`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

- `PS2_Setting`

  Check areas before setting and start setting selected process if all is OK.

  Accepted requests:
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

Remarks
- The message is available since protocol version *015*.

`area.1`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

`area.2`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

`area.3`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

`area.4`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

`area.5`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.ZONE

direction: *output*

This message is used to initiate the CC session for zone operations.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial and only state of the successfully created CC session is `ZIC_Active`. The status can be read using `statusCC.SESSION` request.

The following actions are available in the only CC session state:

- `ZIC_Active`

  Accepted requests:
  - `fnCC.ZONE_ISOLATE`
  - `fnCC.ZONE_UNISOLATE`
  - `fnCC.ZONE_INHIBIT`
  - `fnCC.ZONE_UNINHIBIT`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and can accept all control methods required for Isolate/Inhibit.

Remarks

- The message is available since protocol version *011*.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.DEVICE

direction: *output*

This message is used to initiate the CC session for devices operations.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial and only state of the successfully created CC session is `DIC_Active`. The status can be read using `statusCC.SESSION` request.

The following actions are available in the only CC session state:

- `DIC_Active`

  Accepted requests:
  - `fnCC.DEVICE_ISOLATE`
  - `fnCC.DEVICE_UNISOLATE`
  - `fnCC.BATTERY_TEST_START`
  - `fnCC.BATTERY_TEST_CANCEL`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and can accept all control methods required for Isolate/Inhibit or battery test.

Remarks

- The message is available since protocol version *011*.
- Device operations are limited to Keypads and Expanders only.
- Battery test operations are limited to Expanders only, and are available since protocol version*023*.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.OUT_TRIG

direction: *output*

This message is used to initiate the CC session for output/trigger operations.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial and only state of the successfully created CC session is `OC_Active`. The status can be read using `statusCC.SESSION` request.

The following actions are available in the only CC session state:

- `OC_Active`

  Accepted requests:
  - `fnCC.OUT_TRIG_ACTIVATE`
  - `fnCC.OUT_TRIG_DEACTIVATE`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and can accept all control methods required for changing trigger state.

Remarks

- The message is available since protocol version *011*.
- Only outputs that have filter with Trigger and Remote Output Control event flag can be controlled remotely. Output control function in fact controls only Trigger event flag and it is not possible to update output state directly. It means that not all outputs can be controlled remotely (i.e. you can not switch off the siren using this mechanism).

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.OUTPUT

direction: *output*

This message is used to initiate the CC session for output operations.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial and only state of the successfully created CC session is `OT_Active`. The status can be read using `statusCC.SESSION` request.

The following actions are available in the only CC session state:

- `OT_Active`

  Accepted requests:
  - `fnCC.OUTPUT_ACTIVATE`
  - `fnCC.OUTPUT_DEACTIVATE`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and can accept all control methods required for changing output state.

Remarks

- The message is available since protocol version *011*.
- Only Installer can control outputs using this mechanism.
- It is required to create C&C control object with all areas.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.ENG_RES

direction: *output*

This message is used to initiate the CC session for engineering reset.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial state of the successfully created CC session is `ERC_Ready`. The inital status can be read using `statusCC.SESSION` request.

Depending on the CC session state the following actions are possible:

- `ERC_Ready`

  Accepted requests:
  - `fnCC.ENG_RES_GETCODE`
  - `fnCC.ENG_RES_DORESET`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and can accept engineer reset method.

- `ERC_Finished`

  Accepted requests:
  - `fnCC.ENG_RES_GETRESULT`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Engineer reset process finished. Engineer reset process result can be checked by `fnCC.ENG_RES_GETRESULT` request.

Remarks

- The message is available since protocol version *011*.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.TIME_DATE

direction: *output*

This message is used to initiate the CC session for time/date changes.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial and only state of the successfully created CC session is `TDC_Active`. The status can be read using `statusCC.SESSION` request.

The following actions are available in the only CC session state:

- `TDC_Active`

  Accepted requests:
  - `fnCC.TIME_DATE_SET`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and can accept time/date set request.

Remarks

- The message is available since protocol version *011*.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.PC_CONN

direction: *output*

This message is used to initiate the CC session for PC connection starting.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial and only state of the successfully created CC session is `PCSC_Active`. The status can be read using `statusCC.SESSION` request.

The following actions are available in the only CC session state:

- `TDC_Active`

  Accepted requests:
  - `fnCC.PC_CONN_START`
  - `fnCC.PC_CONN_STOP`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and can accept PCC start method.

Remarks

- The message is available since protocol version *011*.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.USER

direction: *output*

This message is used to initiate the CC session for User.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial and only state of the successfully created CC session is `UC_Active`. The status can be read using `statusCC.SESSION` request.

The following actions are available in the only CC session state:

- `UC_Active`

  Accepted requests:
  - `fnCC.USER_SETCONTROL`
  - `fnCC.USER_SETREPORT`
  - `fnCC.USER_GETPHONE`
  - `fnCC.USER_SETPHONE`
  - `fnCC.USER_SETPIN`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and can accept user control requests.

Remarks

- The message is available since protocol version *011*.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.TEST_CALL

direction: *output*

This message is used to initiate the CC session for CS test call.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial state of the successfully created CC session is `CST_Ready`. The inital status can be read using `statusCC.SESSION` request.

Depending on the CC session state the following actions are possible:

- `CST_Ready`

  Accepted requests:
  - `fnCC.TEST_CALL_START`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and can accept `fnCC.TEST_CALL_START` request, which changes the status to `CST_Active`.

- `CST_Active`

  Accepted requests:
  - `fnCC.TEST_CALL_STATUS`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  CS Test process is in progress. CS Test process state can be checked by `fnCC.TEST_CALL_STATUS` request.

- `CST_Finished`

  Accepted requests:
  - `fnCC.TEST_CALL_STATUS`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  CS Test process is finished. Final CS Test process state can be checked by `fnCC.TEST_CALL_STATUS` request.

Remarks

- The message is available since protocol version *011*.

`area.1`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

`area.2`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

`area.3`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

`area.4`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

`area.5`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

`area.6`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.PICTURES

direction: *output*

This message is used to initiate the CC session for pictures.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.CAM_RANGETST

direction: *output*

This message is used to initiate the CC session for cameras' range test.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial state of the successfully created CC session is `CRT_Ready`. The inital status can be read using `statusCC.SESSION` request.

Depending on the CC session state the following actions are possible:

- `CRT_Ready`

  Accepted requests:
  - `fnCC.CAM_RANGETST_START`
  - `fnCC.CAM_RANGETST_ADDCAM`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and ready to start. It is possible to add a camera to the list of cameras that will be taken into account during range test.

- `CRT_Active`

  Accepted requests:
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Range test process is active. In this state change od state events with range test data for each camera will be posted by panel until the session is terminated by user. When it happens or should there a time-out or other type of error occur, state is changed to `CRT_Finished`.

- `CRT_Finished`

  Accepted requests:
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Range test process finished.

Remarks

- The message is available since protocol version *021*.

`area.1`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

`area.2`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

`area.3`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

`area.4`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

`area.5`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

`area.6`

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.SYS_INV_WALKTST

direction: *output*

This message is used to initiated CC for inverted walk test.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial state of the successfully created CC session is `CRT_Ready`. The inital status can be read using `statusCC.SESSION` request.

Depending on the CC session state the following actions are possible:

- `CIW_Ready`

  Accepted requests:
  - `fnCC.SYS_INV_WALKTST_REP`
  - `fnCC.SYS_INV_WALKTST_RESET`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and ready to start further action.

- `CIW_Active`

  Accepted requests:
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Inverted walk test process is active. In this state change od state events with inverted walk test status data for each zone will be posted by panel. When all data are transmitted state is changed to `CRT_Finished`.

- `CIW_Finished`

  Accepted requests:
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Inverted walk test process finished.

Remarks

- The message is available since protocol version *023*.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

createCC.SYS_WALKTST_MODE

direction: *output*

This message is used to initiated CC for inverted walk test.

The panel response for the message is `return.short`. The response contains the session ID, which will be used in every further CC functions. To destroy the CC session use `destroyCC.SESSION` request.

The initial and only state of the successfully created CC session is `CWM_Active`. The status can be read using `statusCC.SESSION` request.

The following actions are available in the only CC session state:

- `CWM_Active`

  Accepted requests:
  - `fnCC.SYS_CHANGE_WALKTST_MODE`
  - `statusCC.SESSION`
  - `destroyCC.SESSION`

  Control task is created and can accept user control requests.

Remarks

- The message is available since protocol version *023*.

area.1

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 3.

area.4

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 5.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.6

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 6.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.7

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 7.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

area.8

multiplicity: *single (static)*
type: *boolean*

Indicates whether the session operates on area 8.

Remarks

- The property might be set to `true` only for Ats Advanced models *ATS2000A* or *ATS2000AIP*.

destroyCC.SESSION

direction: *output*

This message is used to destroy the CC session with the specified session ID.

The panel response for the message is `return.void`.

See also

- `createCC.A_STATE`
- `createCC.A_SET`
- `createCC.A_CONFAL`
- `createCC.A_WALKTST`
- `createCC.A_UNSET`
- `createCC.A_PARTSET`
- `createCC.A_PARTSET2`
- `createCC.ZONE`
- `createCC.DEVICE`
- `createCC.OUT_TRIG`
- `createCC.OUTPUT`
- `createCC.ENG_RES`
- `createCC.TIME_DATE`
- `createCC.PC_CONN`
- `createCC.USER`
- `createCC.TEST_CALL`
- `createCC.PICTURES`
- `createCC.CAM_RANGETST`
- `createCC.SYS_INV_WALKTST`
- `createCC.SYS_WALKTST_MODE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

statusCC.SESSION

direction: *output*

This message is used to get the status of CC session with the specified session ID.

The panel response for the message is `return.statusCC`. The returned value is the current state ID.

See also

- `createCC.A_STATE`
- `createCC.A_SET`
- `createCC.A_CONFAL`
- `createCC.A_WALKTST`
- `createCC.A_UNSET`
- `createCC.A_PARTSET`
- `createCC.A_PARTSET2`
- `createCC.ZONE`
- `createCC.DEVICE`
- `createCC.OUT_TRIG`
- `createCC.OUTPUT`
- `createCC.ENG_RES`
- `createCC.TIME_DATE`
- `createCC.PC_CONN`
- `createCC.USER`
- `createCC.TEST_CALL`
- `createCC.CAM_RANGETST`
- `createCC.SYS_INV_WALKTST`
- `createCC.SYS_WALKTST_MODE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

return.statusCC

direction: *input*

This is the return message containing command and control operational state ID .

See also

- statusCC.SESSION
- fnCC.CAM_RANGETST_START
- fnCC.CAM_RANGETST_ADDCAM
- fnCC.A_STATE_GET_INH
- fnCC.A_STATE_GET_UNINH
- fnCC.A_SET_SETAREAS
- fnCC.A_SET_GETFAULT
- fnCC.A_SET_GETACTIVE
- fnCC.A_SET_GETINHIB
- fnCC.A_SET_INHFAULT
- fnCC.A_SET_INHACTIVE
- fnCC.A_SET_FORCEDSET
- fnCC.A_CONFAL_START
- fnCC.A_CONFAL_GETALARM
- fnCC.A_CONFAL_CONFALARM
- fnCC.A_WALKTST_START
- fnCC.A_WALKTST_GETLIST
- fnCC.A_WALKTST_GETEV
- fnCC.A_WALKTST_GETRES
- fnCC.A_WALKTST_GET_WARN_TIME
- fnCC.SYS_INV_WALKTST_REP
- fnCC.SYS_INV_WALKTST_RESET
- fnCC.SYS_CHANGE_WALKTST_MODE
- fnCC.A_WALKTST_START_WITH_REP
- fnCC.A_WALKTST_ADD_ZONE
- fnCC.A_UNSET_UNSETAREAS
- fnCC.A_UNSET_SKIP
- fnCC.A_UNSET_GETALARM
- fnCC.A_UNSET_CONFALARM
- fnCC.A_UNSET_GETFAULT
- fnCC.A_UNSET_CONFFAULT
- fnCC.ENG_RES_DORESET
- fnCC.ENG_RES_GETRESULT
- fnCC.ENG_RES_GETCODE
- fnCC.TEST_CALL_START
- fnCC.TEST_CALL_STATUS

stateID

multiplicity: *single (static)*
type: *integer*

Current state ID for command and control operation (16 bit).

value	symbol
#0000	CSMS_UNKNOWN
#0001	CSMS_DESTROYED
#0002	CSMS_CREATED
#0100	CSMS_CC_Ready
#0101	CSMS_CC_CnfAlarms
#0102	CSMS_CC_Confirmed
#0200	CSMS_WC_Ready
#0201	CSMS_WC_Active
#0202	CSMS_WC_Finished
#0300	CSMS_UC_Ready
#0301	CSMS_UC_Unsetting
#0302	CSMS_UC_CnfAlarms
#0303	CSMS_UC_CnfFaults
#0304	CSMS_UC_Unset
#0400	CSMS_PC_Ready
#0401	CSMS_PC_Faults
#0402	CSMS_PC_ActiveStates
#0403	CSMS_PC_Inhibited
#0404	CSMS_PC_Setting
#0405	CSMS_PC_Set
#0500	CSMS_FC_Ready
#0501	CSMS_FC_Faults
#0502	CSMS_FC_ActiveStates
#0503	CSMS_FC_Inhibited
#0504	CSMS_FC_Setting
#0505	CSMS_FC_Set
#0600	CSMS_SC_Active
#0700	CSMS_TC_Active
#0701	CSMS_TC_Ready
#0702	CSMS_TC_Finished
#0800	CSMS_ZIC_Active
#0900	CSMS_DIC_Active
#0A00	CSMS_OC_Active
#0B00	CSMS_ERC_Ready
#0B01	CSMS_ERC_Finished
#0C00	CSMS_TDC_Active
#0D00	CSMS_PCSC_Active
#0E00	CSMS_Out_Active
#0F00	CSMS_UC_Active

value	symbol
#1000	CSMS_PC2_Ready
#1001	CSMS_PC2_Faults
#1002	CSMS_PC2_ActiveStates
#1003	CSMS_PC2_Inhibited
#1004	CSMS_PC2_Setting
#1005	CSMS_PC2_Set
#1100	CSMS_PICTURES_Ready
#1101	CSMS_PICTURES_Prepared
#1102	CSMS_PICTURES_Active
#1200	CSMS_RT_Ready
#1201	CSMS_RT_Active
#1202	CSMS_RT_Finished
#1300	CSMS_IW_Ready
#1301	CSMS_IW_Active
#1302	CSMS_IW_Finished
#1400	CSMS_WM_Active

fnCC.CAM_RANGETST_START

direction: *output*

This message is used to start range test for the current CC session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *021*.

See also
- `createCC.CAM_RANGETST`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.CAM_RANGETST_ADDCAM

direction: *output*

This message is used to add a camera to the range test camera list.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *021*.

See also
- `createCC.CAM_RANGETST`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.A_STATE_GET_INH

direction: *output*

This message is used to get the events that make areas not ready to set, but can be inhibited by user.

The following responses are possible:
- `return.sysevent` — The system event data are received.
- `return.void` — There is no more data available.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *021*.

See also
- `createCC.A_STATE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

fnCC.A_STATE_GET_UNINH

direction: *output*

This message is used to get the events that make areas not ready to set and cannot be inhibited by user.

The following responses are possible:
- `return.sysevent` — The system event data are received.
- `return.void` — There is no more data available.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *021*.

See also
- `createCC.A_STATE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

fnCC.A_SET_SETAREAS

direction: *output*

This message is used to proceed with arming the areas for the current CC session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_SET`
- `createCC.A_PARTSET`
- `createCC.A_PARTSET2`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.A_SET_GETFAULT

direction: *output*

This message is used to get the faults for the current CC arming session.

The following responses are possible:
- `return.sysevent` — The fault data are received.
- `return.void` — There is no more data available.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_SET`
- `createCC.A_PARTSET`
- `createCC.A_PARTSET2`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

fnCC.A_SET_GETACTIVE

direction: *output*

This message is used to get the active states for the current CC arming session.

The following responses are possible:
- `return.sysevent` — The active state data are received.
- `return.void` — There is no more data available.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_SET`
- `createCC.A_PARTSET`
- `createCC.A_PARTSET2`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

fnCC.A_SET_GETINHIB

direction: *output*

This message is used to get the inhibited/isolated zones/devices for the current CC arming session.

The following responses are possible:
- `return.sysevent` — The inhibited/isolated zone/device data are received.
- `return.void` — There is no more data available.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_SET`
- `createCC.A_PARTSET`
- `createCC.A_PARTSET2`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

fnCC.A_SET_INHFAULT

direction: *output*

This message is used to inhibit the specified fault for the current CC arming session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_SET`
- `createCC.A_PARTSET`
- `createCC.A_PARTSET2`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

eventUniqueID

multiplicity: *single (static)*
type: *integer*

Event unique identifier (16 bit).

fnCC.A_SET_INHACTIVE

direction: *output*

This message is used to inhibit the specified active state for the current CC arming session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_SET`
- `createCC.A_PARTSET`
- `createCC.A_PARTSET2`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

eventUniqueID

multiplicity: *single (static)*
type: *integer*

Event unique identifier (16 bit).

fnCC.A_SET_FORCEDSET

direction: *output*

This message is used to force arming of the areas for the current CC session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_SET`
- `createCC.A_PARTSET`
- `createCC.A_PARTSET2`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.A_CONFAL_START

direction: *output*

This message is used to start alarm confirming for the current CC session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_CONFAL`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.A_CONFAL_GETALARM

direction: *output*

This message is used to get the alarm to confirm for the current CC arming session.

The following responses are possible:
- `return.sysevent` — The alarm data are received.
- `return.void` — There is no more data available.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_CONFAL`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

fnCC.A_CONFAL_CONFALARM

direction: *output*

This message is used to confirm the specified alarm for the current CC arming session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_CONFAL`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

eventUniqueID

multiplicity: *single (static)*
type: *integer*

Event unique identifier (16 bit).

fnCC.A_WALKTST_START

direction: *output*

This message is used to start walk test for the current CC session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_WALKTST`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.A_WALKTST_GETLIST

direction: *output*

This message is used to get the events to be tested during the walk test for the current CC session.

The following responses are possible:
- `return.sysevent` — The alarm data are received.
- `return.void` — There is no more data available.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_WALKTST`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

fnCC.A_WALKTST_GETEV

direction: *output*

This message is used to get the walk test events which has already been tested for the current CC session.

The following responses are possible:
- `return.sysevent` — The alarm data are received.
- `return.void` — There is no more data available.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_WALKTST`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

fnCC.A_WALKTST_GETRES

direction: *output*

This message is used to get the results of walk test for the current CC arming session.

The following responses are possible:
- `return.sysevent` — The alarm data are received.
- `return.void` — There is no more data available.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_WALKTST`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

fnCC.A_WALKTST_GET_WARN_TIME

direction: *output*

This message is used to get PIR settle warning time value.

The following responses are possible:
- `return.short` — Time in seconds. Value 0 menas that there is no PIRCAM zone to walktest.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *022*.

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.SYS_INV_WALKTST_REP

direction: *output*

This message is used to start report inverted walk test for the current CC session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *023*.

See also
- `createCC.SYS_INV_WALKTST`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.SYS_INV_WALKTST_RESET

direction: *output*

This message is used to reset inverted walk test for the current CC session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *023*.

See also
- `createCC.SYS_INV_WALKTST`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.SYS_CHANGE_WALKTST_MODE

direction: *output*

This message is used to change walk test mode.

The following responses are possible:
- `return.bool` — The flag indicating weather the walk test mode has changed.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *023*.

See also
- `createCC.SYS_WALKTST_MODE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

mode

multiplicity: *single (static)*
type: *integer*

Walk test mode change type.

value	symbol
0	Clear
1	Set
2	Toggle

fnCC.A_WALKTST_START_WITH_REP

direction: *output*

This message is used to start walk test with reporting for the current CC session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *023*.

See also
- `createCC.A_WALKTST`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.A_WALKTST_ADD_ZONE

direction: *output*

This message is used to manually add zone in walk test for the current CC session.

The following responses are possible:
- `return.bool` — The value indicating whether the zone is succeessfully added to the walk test list.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *023*.

See also
- `createCC.A_WALKTST`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

zone

multiplicity: *single (static)*
type: *integer*

Zone index for walk test per zone.

fnCC.A_UNSET_UNSETAREAS

direction: *output*

This message is used to disarm the areas for the current CC session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_UNSET`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.A_UNSET_SKIP

direction: *output*

This message is used to skip alarm/fault confirming phase for the current CC session.

The following responses are possible:
- `return.void` — Confirmation.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

See also
- `createCC.A_UNSET`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.A_UNSET_GETALARM

direction: *output*

This message is used to get the alarm to confirm for the current CC disarming session.

The following responses are possible:
- `return.sysevent` — The alarm data are received.
- `return.void` — There is no more data available.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_UNSET`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

fnCC.A_UNSET_CONFALARM

direction: *output*

This message is used to confirm the specified alarm for the current CC disarming session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_UNSET`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

eventUniqueID

multiplicity: *single (static)*
type: *integer*

Event unique identifier (16 bit).

fnCC.A_UNSET_GETFAULT

direction: *output*

This message is used to get the fault to confirm for the current CC disarming session.

The following responses are possible:
- `return.sysevent` — The alarm data are received.
- `return.void` — There is no more data available.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_UNSET`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

next

multiplicity: *single (static)*
type: *integer*

Event to get (FALSE - the first, TRUE - the next).

value	symbol
0	false
1	true

fnCC.A_UNSET_CONFFAULT

direction: *output*

This message is used to confirm the specified fault for the current CC disarming session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.A_UNSET`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

eventUniqueID

multiplicity: *single (static)*
type: *integer*

Event unique identifier (16 bit).

fnCC.ZONE_ISOLATE

direction: *output*

This message is used to isolate the specified zone for the current CC session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.ZONE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.ZONE_UNISOLATE

direction: *output*

This message is used to unisolate the specified zone for the current CC session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.ZONE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.ZONE_INHIBIT

direction: *output*

This message is used to inhibit the specified zone for the current CC session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.ZONE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.ZONE_UNINHIBIT

direction: *output*

This message is used to uninhibit the specified zone for the current CC session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.ZONE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.DEVICE_ISOLATE

direction: *output*

This message is used to isolate the specified device for the current CC session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.
- Device isolate operations are limited to Keypads and Expanders only.

See also

- `createCC.DEVICE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

classID

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Device class identifier (16 bit).

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC

value	symbol
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

deviceID

multiplicity: *single (static)*
type: *integer*

Device identifier (16 bit).

fnCC.DEVICE_UNISOLATE

direction: *output*

This message is used to unisolate the specified device for the current CC session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.
- Device unisolate operations are limited to Keypads and Expanders only.

See also

- `createCC.DEVICE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

classID

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Device class identifier (16 bit).

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC

value	symbol
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

deviceID

multiplicity: *single (static)*
type: *integer*

Device identifier (16 bit).

fnCC.BATTERY_TEST_START

direction: *output*

This message is used to start battery test.

The response is `return.bool` — the operation status.

Remarks
- The message is available since protocol version *023*.
- Battery test operations are limited to expanders only.

See also
- `createCC.DEVICE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

classID

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Device class identifier (16 bit).

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC

value	symbol
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

deviceID

multiplicity: *single (static)*
type: *integer*

Device identifier (16 bit).

fnCC.BATTERY_TEST_CANCEL

direction: *output*

This message is used to cancel battery test.

The response is `return.bool` — the operation status.

Remarks
- The message is available since protocol version *023*.
- Battery test operations are limited to expanders only.

See also
- `createCC.DEVICE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

classID

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Device class identifier (16 bit).

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC

value	symbol
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

deviceID

multiplicity: *single (static)*
type: *integer*

Device identifier (16 bit).

fnCC.OUT_TRIG_ACTIVATE

direction: *output*

This message is used to activate the specified trigger object for the current CC session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.OUT_TRIG`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.OUT_TRIG_DEACTIVATE

direction: *output*

This message is used to deactivate the specified trigger object for the current CC session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.OUT_TRIG`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.OUTPUT_ACTIVATE

direction: *output*

This message is used to activate the specified output object for the current CC session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.OUTPUT`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.OUTPUT_DEACTIVATE

direction: *output*

This message is used to deactivate the specified output object for the current CC session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.OUTPUT`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.ENG_RES_DORESET

direction: *output*

This message is used to perform engineer reset for the current CC session.

The following responses are possible:
- `return.void` — OK confirmation.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.ENG_RES`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

resCode

multiplicity: *single (static)*
type: *integer*

Reset code (16 bit).

fnCC.ENG_RES_GETRESULT

direction: *output*

This message is used to get the result of engineer reset operation for the current CC session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.ENG_RES`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.ENG_RES_GETCODE

direction: *output*

This message is used to get the engineer reset code.

The following responses are possible:
- `return.short` — The engineer reset code.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.ENG_RES`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

fnCC.TIME_DATE_SET

direction: *output*

This message is used to set the panel time and date.

The response is `return.bool` — the operation status.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.TIME_DATE`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

timeStamp

multiplicity: *single (static)*
type: *datetime*

The new timestamp value.

Remarks
- The timestamp value must be rounded to seconds precision.
- It is assumed that the value uses Coordinated Universal Time format aka *UTC*.
- Leap seconds available in *UTC* are not supported.

format: `date+time`

fnCC.PC_CONN_START

direction: *output*

This message is used to start PC connection session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.PC_CONN`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.PC_CONN_STOP

direction: *output*

This message is used to stop PC connection session.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.PC_CONN`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.USER_SETCONTROL

direction: *output*

This message is used to set User SMS control flag.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.USER`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

userID

multiplicity: *single (static)*
type: *integer*

User identifier/index (16 bit).

smsControl

multiplicity: *single (static)*
type: *integer*

SMS control enable/disable flag.

value	symbol
0	false
1	true

fnCC.USER_SETREPORT

direction: *output*

This message is used to set User SMS reporting flag.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.USER`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

userID

multiplicity: *single (static)*
type: *integer*

User identifier/index (16 bit).

smsReporting

multiplicity: *single (static)*
type: *integer*

Sms Reporting mode.

value	symbol
0	USRM_OFF
1	USRM_ON
2	USRM_REARM

fnCC.USER_GETPHONE

direction: *output*

This message is used to get User phone number.

The response is `return.UserPhone` — the phone number of the user.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.USER`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

userID

multiplicity: *single (static)*
type: *integer*

User identifier/index (16 bit).

return.UserPhone

direction: *input*

This message returns User phone number.

Remarks

- The message is available since protocol version *011*.

See also

- fnCC.USER_GETPHONE

userPhone

multiplicity: *single (static)*
type: *string*

The phone number.

The maximum length of the phone number string is 20 characters. The valid characters are 0123456789 (codes between 0x30 and 0x39), T (code 0x54) and P (code 0x50).

fnCC.USER_SETPHONE

direction: *output*

This message is used to set User phone number.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *011*.

See also

- `createCC.USER`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

userID

multiplicity: *single (static)*
type: *integer*

User identifier/index (16 bit).

userPhone

multiplicity: *single (static)*
type: *string*

The phone number.

The maximum length of the phone number string is 20 characters. The valid characters are `0123456789` (codes between `0x30` and `0x39`), `T` (code `0x54`) and `P` (code `0x50`).

fnCC.USER_SETPIN

direction: *output*

This message is used to set User PIN.

The response is `return.bool` — the operation status.

Remarks

- The message is available since protocol version *023*.

See also

- `createCC.USER`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

userID

multiplicity: *single (static)*
type: *integer*

User identifier/index (16 bit).

userPIN

multiplicity: *single (static)*
type: *string*

The PIN number.

The maximum length of the PIN number string is 10 characters. The valid characters are `0123456789` (codes between `0x30` and `0x39`), `T` (code `0x54`) and `P` (code `0x50`).

fnCC.OUT_SCHED_TRIG_ACTIVATE

direction: *output*

This message is used to activate the specified schedule trigger object for the current CC session.

The response is return.bool (operation status)

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.OUT_SCHED_TRIG_DEACTIVATE

direction: *output*

This message is used to deactivate the specified schedule trigger object for the current CC session.

The response is return.bool (operation status)

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.TEST_CALL_START

direction: *output*

This message is used to start CS test call session.

The following responses are possible:
- `return.bool` — The operation status.
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.TEST_CALL`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

objectID

multiplicity: *single (static)*
type: *integer*

Object identifier/index (16 bit).

fnCC.TEST_CALL_STATUS

direction: *output*

This message is used to get the CS test call session status.

The following responses are possible:
- `return.short` — The CS Test state (see below).
- `return.statusCC` — The state has changed and the operation is not allowed in the current state.

The CS Test states:

Value	Finished	Description
1	NO	CS is ready
2	YES	Line fault on the path attached to the CS.
3	YES	No dial tone detected.
4	YES	No handshake detected.
5	YES	No kissoff detected.
6	YES	Call successfull.
7	NO	Acknowledge received from remote station.
8	NO	Handshake received from remote station.
9	NO	Dialing/connecting to the remote station.
10	YES	CS in error state.

Remarks
- The message is available since protocol version *011*.

See also
- `createCC.TEST_CALL`

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

select.ZoneNames

direction: *output*

This is the outgoing call to select zone names.

The response consists of up to 16 zone names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Zone index.

min	max
1	128
129	256
257	368

return.ZoneNames

direction: *input*

This is the response for , which consists of up to 16 zone names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Zone index.

min	max
1	128
129	256
257	368

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.AreaNames

direction: *output*

This is the outgoing call to select area names.

The response consists of up to 16 area names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Area index.

min	max
1	8

return.AreaNames

direction: *input*

This is the response for , which consists of up to 16 area names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Area index.

min	max
1	8

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.RASNames

direction: *output*

This is the outgoing call to select keypad names.

The response consists of up to 16 keypad names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

return.RASNames

direction: *input*

This is the response for , which consists of up to 16 ras names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.DGPNames

direction: *output*

This is the outgoing call to select expander names.

The response consists of up to 16 expander names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

DGP index.

min	max
1	7

return.DGPNames

direction: *input*

This is the response for , which consists of up to 16 expander names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

DGP index.

min	max
1	7

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.UserNames

direction: *output*

This is the outgoing call to select user names.

The response consists of up to 16 user names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

return.UserNames

direction: *input*

This is the response for , which consists of up to 16 user names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.OutputNames

direction: *output*

This is the outgoing call to select output names.

The response consists of up to 16 output names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

OUTPUT index.

min	max
1	200

return.OutputNames

direction: *input*

This is the response for , which consists of up to 16 output names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

OUTPUT index.

min	max
1	200

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.CEvFilterNames

direction: *output*

This is the outgoing call to select conditional filter names.

The response consists of up to 16 conditional filter names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

CEvFilter index.

min	max
1	64

return.CEvFilterNames

direction: *input*

This is the response for , which consists of up to 16 conditional filter names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

CEvFilter index.

min	max
1	64

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.UserGroupNames

direction: *output*

This is the outgoing call to select user group names.

The response consists of up to 16 user group names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

UserGroup index.

min	max
1	16

return.UserGroupNames

direction: *input*

This is the response for , which consists of up to 16 user group names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

UserGroup index.

min	max
1	16

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.CSNames

direction: *output*

This is the outgoing call to select central station names.

The response consists of up to 16 central station names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

return.CSNames

direction: *input*

This is the response for , which consists of up to 16 central station names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.DLNames

direction: *output*

This is the outgoing call to select zone names.

The response consists of up to 16 zone names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

return.DLNames

direction: *input*

This is the response for , which consists of up to 16 zone names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.SYSNames

direction: *output*

This is the outgoing call to select system name.

The response consists system name.

Remarks

- The message is available since protocol version *021*.

return.SYSNames

direction: *input*

This is the response for , which consists the system name.

Remarks

- The message is available since protocol version *021*.
- Due to compatybility reasons with the other messages of the same kind the name is available as dynamic property at index 1.

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.PCCNames

direction: *output*

This is the outgoing call to select PC connection names.

The response consists of up to 16 PC connection names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

return.PCCNames

direction: *input*

This is the response for , which consists of up to 16 PC connection names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.TriggerNames

direction: *output*

This is the outgoing call to select trigger names.

The response consists of up to 16 trigger names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Trigger index.

min	max
1	255

return.TriggerNames

direction: *input*

This is the response for , which consists of up to 16 trigger names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Trigger index.

min	max
1	255

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.SchedActNames

direction: *output*

This is the outgoing call to select schedule action names.

The response consists of up to 16 schedule action names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

return.SchedActNames

direction: *input*

This is the response for , which consists of up to 16 schedule action names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.SchedActLstNames

direction: *output*

This is the outgoing call to select schedule action list names.

The response consists of up to 16 schedule action list names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Schedule action lists index.

min	max
1	32

return.SchedActLstNames

direction: *input*

This is the response for , which consists of up to 16 schedule action list names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Schedule action lists index.

min	max
1	32

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.SchedExcNames

direction: *output*

This is the outgoing call to select schedule exception names.

The response consists of up to 16 schedule exception names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Schedule exceptions index.

min	max
1	64

return.SchedExcNames

direction: *input*

This is the response for , which consists of up to 16 schedule exception names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Schedule exceptions index.

min	max
1	64

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.ScheduleNames

direction: *output*

This is the outgoing call to select schedule names.

The response consists of up to 16 schedule names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Schedules index.

min	max
1	4

return.ScheduleNames

direction: *input*

This is the response for , which consists of up to 16 schedule names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Schedules index.

min	max
1	4

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.FobNames

direction: *output*

This is the outgoing call to select fob names.

The response consists of up to 16 fob names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

return.FobNames

direction: *input*

This is the response for , which consists of up to 16 fob names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.CameraNames

direction: *output*

This is the outgoing call to select camera names.

The response consists of up to 16 camera names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *021*.

index

multiplicity: *single (static)*
type: *integer*

Camera index.

min	max
1	368

return.CameraNames

direction: *input*

This is the response for , which consists of up to 16 camera names starting from the position selected in the `index` property.

Remarks

- The message is available since protocol version *022*.

index

multiplicity: *single (static)*
type: *integer*

Camera index.

min	max
1	368

name

multiplicity: *multiple (dynamic)*
type: *string*

Name of the object

select.Zone

direction: *output*

This is the outgoing call for selectZone method.

index

multiplicity: *single (static)*
type: *integer*

Zone index.

min	max
1	128
129	256
257	368

insert.Zone

direction: *output*

This is the outgoing call for "insertZone" method.

index

multiplicity: *single (static)*
type: *integer*

Zone index.

min	max
1	128
257	368

name

multiplicity: *single (static)*
type: *string*

Zone name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

type

multiplicity: *single (static)*
type: *integer*

value	symbol
1	alarm
2	entry-exit
3	access
4	fire
5	panic
6	24h
7	tamper
8	exitTerminator
9	keyswitch
10	medical
11	technical
12	txPathFault
13	firedoor
14	auxMainsFail
15	auxBattFail
16	keybox
17	engReset
18	entry-exit-2

area.1

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 2.

area.4

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 2.

area.5

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 2.

area.6

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 2.

area.7

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 2.

area.8

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 2.

partSet

multiplicity: *single (static)*
type: *integer*

Part set property.

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can be changed.
entry-exit	*false*	Property can be changed.
access	*false*	Property can be changed.

Zone type	Default value	Restriction
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can be changed.
keyswitch	*false*	Property can be changed.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can be changed.

value	symbol
0	false
1	true

doorBell

multiplicity: *single (static)*
type: *integer*

Door bell property.

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*off*	Property can be changed. Check *doorBell* type for details.
entry-exit	*off*	Property can be changed. Check *doorBell* type for details.
access	*off*	Property can be changed. Check *doorBell* type for details.
fire	*off*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*off*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*off*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*off*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*off*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*off*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*off*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*off*	Property can be changed for this type of zone since protocol version 019. Otherwise property can NOT be changed and default value must be send.

Zone type	Default value	Restriction
txPathFault	*off*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*off*	Property can be changed. Check *doorBell* type for details.
auxMainsFail	*off*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*off*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*off*	Property can be changed. Check *doorBell* type for details.

value	symbol
0	off
1	buzzer
2	siren

dualInput

multiplicity: *single (static)*
type: *integer*

Alarm is generated if two zones in area are active.

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can be changed.
entry-exit	*false*	Property can NOT be changed for this type of zone. Default value must be send.
access	*false*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*false*	Property can be changed.
panic	*false*	Property can be changed.
24h	*false*	Property can be changed.

Zone type	Default value	Restriction
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can be changed.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can be changed.
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can NOT be changed for this type of zone. Default value must be send.

value	symbol
0	false
1	true

antiMask

multiplicity: *single (static)*
type: *integer*

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can be changed.
entry-exit	*false*	Property can be changed.

Zone type	Default value	Restriction
access	*false*	Property can be changed.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can be changed.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can be changed.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can be changed.
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can be changed.

value	symbol
0	false
1	true

soakTest

multiplicity: *single (static)*
type: *integer*

Soak test property.

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can be changed.
entry-exit	*false*	Property can be changed.
access	*false*	Property can be changed.
fire	*false*	Property can be changed.
panic	*false*	Property can be changed.
24h	*false*	Property can be changed.
tamper	*false*	Property can be changed.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can be changed.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can be changed.
auxMainsFail	*false*	Property can be changed.
auxBattFail	*false*	Property can be changed.
entry-exit-2	*false*	Property can be changed.

value	symbol
0	false
1	true

doubleKnock

multiplicity: *single (static)*
type: *integer*

Double knock

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can be changed.
entry-exit	*false*	Property can NOT be changed for this type of zone. Default value must be send.
access	*false*	Property can be changed.
fire	*false*	Property can be changed.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can be changed.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can be changed.
txPathFault	*false*	Property can be changed.
firedoor	*false*	Property can be changed.
auxMainsFail	*false*	Property can be changed.
auxBattFail	*false*	Property can be changed.
entry-exit-2	*false*	Property can NOT be changed for this type of zone. Default value must be send.

The option is mutually exclusive with *Zone Held Open* option.

value	symbol
0	false
1	true

userWalktest

multiplicity: *single (static)*
type: *integer*

User walktest

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*true*	Property can be changed.
entry-exit	*true*	Property can be changed.
access	*true*	Property can be changed.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*true*	Property can be changed.
24h	*true*	Property can be changed.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*true*	Property can be changed.
technical	*false*	Property can be changed.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*true*	Property can be changed.
auxMainsFail	*false*	Property can be changed.

Zone type	Default value	Restriction
auxBattFail	*false*	Property can be changed.
entry-exit-2	*true*	Property can be changed.

value	symbol
0	false
1	true

engWalktest

multiplicity: *single (static)*
type: *integer*

Engineer walktest

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can be changed.
entry-exit	*false*	Property can be changed.
access	*false*	Property can be changed.
fire	*true*	Property can be changed.
panic	*true*	Property can be changed.
24h	*false*	Property can be changed.
tamper	*false*	Property can be changed.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*true*	Property can be changed.
technical	*true*	Property can be changed.
txPathFault	*true*	Property can be changed.
firedoor	*true*	Property can be changed.
auxMainsFail	*true*	Property can be changed.
auxBattFail	*true*	Property can be changed.
entry-exit-2	*false*	Property can be changed.

value	symbol
0	false
1	true

setKey

multiplicity: *single (static)*
type: *integer*

Set key

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*off*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*off*	Property can NOT be changed for this type of zone. Default value must be send.
access	*off*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*off*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*off*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*off*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*off*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*off*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*off*	Property can be changed. Check *setKey* values.

Zone type	Default value	Restriction
medical	*off*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*off*	Property can NOT be changed for this type of zone. Default value must be send.
txPathFault	*off*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*off*	Property can NOT be changed for this type of zone. Default value must be send.
auxMainsFail	*off*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*off*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*off*	Property can NOT be changed for this type of zone. Default value must be send.

value	symbol
0	off
1	partSet
2	fullSet
4	partSet2

unSetKey

multiplicity: *single (static)*
type: *integer*

Unset key

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
entry-exit	*false*	Property can NOT be changed for this type of zone. Default value must be send.
access	*false*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can be changed.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can NOT be changed for this type of zone. Default value must be send.

value	symbol
0	false
1	true

latchKey

multiplicity: *single (static)*
type: *integer*

Latch key

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*false*	Property can NOT be changed for this type of zone. Default value must be send.
access	*false*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can be changed.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can NOT be changed for this type of zone. Default value must be send.

value	symbol
0	false
1	true

techPartSet

multiplicity: *single (static)*
type: *integer*

Technical partset

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*false*	Property can NOT be changed for this type of zone. Default value must be send.
access	*false*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*true*	Property can be changed.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can NOT be changed for this type of zone. Default value must be send.

value	symbol
0	false
1	true

techFullSet

multiplicity: *single (static)*
type: *integer*

Technical fullset

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*false*	Property can NOT be changed for this type of zone. Default value must be send.
access	*false*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*true*	Property can be changed.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can NOT be changed for this type of zone. Default value must be send.

value	symbol
0	false
1	true

techUnset

multiplicity: *single (static)*
type: *integer*

Technical unset

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*false*	Property can NOT be changed for this type of zone. Default value must be send.
access	*false*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*true*	Property can be changed.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can NOT be changed for this type of zone. Default value must be send.

value	symbol
0	false
1	true

finalDoorSet

multiplicity: *single (static)*
type: *integer*

Final door set

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*false*	Property can be changed.
access	*false*	Property can be changed.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can be changed.

value	symbol
0	false
1	true

swingerShunt

multiplicity: *single (static)*
type: *integer*

Swinger shunt

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can be changed.

Zone type	Default value	Restriction
entry-exit	*false*	Property can be changed.
access	*false*	Property can be changed.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can be changed.
tamper	*false*	Property can be changed.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can be changed.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can be changed.
auxMainsFail	*false*	Property can be changed.
auxBattFail	*false*	Property can be changed.
entry-exit-2	*false*	Property can be changed.

value	symbol
0	false
1	true

lcd

multiplicity: *single (static)*
type: *integer*

LCD

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*false*	Property can NOT be changed for this type of zone. Default value must be send.
access	*false*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can be changed.
24h	*false*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can be changed.
technical	*false*	Property can be changed.
txPathFault	*false*	Property can be changed.
firedoor	*false*	Property can be changed.
auxMainsFail	*false*	Property can be changed.
auxBattFail	*false*	Property can be changed.
entry-exit-2	*false*	Property can NOT be changed for this type of zone. Default value must be send.

value	symbol
0	false
1	true

eee

multiplicity: *single (static)*
type: *integer*

Extended entry/exit

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*false*	Property can be changed.
access	*false*	Property can be changed.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
firedoor	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can be changed.

value	symbol
0	false
1	true

csReport

multiplicity: *single (static)*
type: *integer*

Report to central station

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*true*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*true*	Property can NOT be changed for this type of zone. Default value must be send.
access	*true*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*true*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*true*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
24h	*true*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*true*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*true*	Property can be changed.
technical	*true*	Property can be changed.
txPathFault	*true*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*true*	Property can NOT be changed for this type of zone. Default value must be send.
auxMainsFail	*true*	Property can be changed.
auxBattFail	*true*	Property can be changed.
entry-exit-2	*true*	Property can NOT be changed for this type of zone. Default value must be send.

value	symbol
0	false
1	true

log

multiplicity: *single (static)*
type: *integer*

Log.

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*true*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*true*	Property can NOT be changed for this type of zone. Default value must be send.
access	*true*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*true*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*true*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*true*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*true*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*true*	Property can be changed.
technical	*true*	Property can be changed.
txPathFault	*true*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*true*	Property can NOT be changed for this type of zone. Default value must be send.
auxMainsFail	*true*	Property can be changed.
auxBattFail	*true*	Property can be changed.

Zone type	Default value	Restriction
entry-exit-2	*true*	Property can NOT be changed for this type of zone. Default value must be send.

value	symbol
0	false
1	true

delayT

multiplicity: *single (static)*
type: *integer*

Delay timer.

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*false*	Property can NOT be changed for this type of zone. Default value must be send.
access	*false*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can be changed.
txPathFault	*false*	Property can be changed.
firedoor	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxMainsFail	*false*	Property can be changed.
auxBattFail	*false*	Property can be changed.
entry-exit-2	*false*	Property can NOT be changed for this type of zone. Default value must be send.

value	symbol
0	false
1	true

shock

multiplicity: *single (static)*
type: *integer*

Shock sensor.

Zone can be set as shock sensor if it has appropriate type (check list below) AND is placed at input 1-8 on panel (zone *index* from 1 to 8) or 1-8 on input expander (zone *index* from 9 to 16).

Zone type	Default value	Restriction
alarm	*false*	Property can be changed.
entry-exit	*false*	Property can be changed.
access	*false*	Property can be changed.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can be changed.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*false*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can be changed.

value	symbol
0	false
1	true

inhibit

multiplicity: *single (static)*
type: *integer*

Inhibit option.

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*true*	Property can be changed.
entry-exit	*true*	Property can be changed.
access	*true*	Property can be changed.
fire	*true*	Property can be changed.
panic	*true*	Property can be changed.
24h	*true*	Property can be changed.
tamper	*true*	Property can be changed.
exitTerminator	*true*	Property can be changed.
keyswitch	*true*	Property can be changed.
medical	*false*	Property can be changed.
technical	*true*	Property can be changed.
txPathFault	*true*	Property can be changed.
firedoor	*true*	Property can be changed.
auxMainsFail	*true*	Property can be changed.
auxBattFail	*true*	Property can be changed.
entry-exit-2	*true*	Property can be changed.

value	symbol
0	false
1	true

pulseCnt

multiplicity: *single (static)*
type: *integer*

Pulse count.

min	max
0	9

grossLvl

multiplicity: *single (static)*
type: *integer*

Gross level.

min	max
0	9

isolate

multiplicity: *single (static)*
type: *integer*

Isolate option.

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can be changed.
entry-exit	*false*	Property can be changed.
access	*false*	Property can be changed.
fire	*false*	Property can be changed.
panic	*false*	Property can be changed.
24h	*false*	Property can be changed.
tamper	*false*	Property can be changed.
exitTerminator	*false*	Property can be changed.
keyswitch	*false*	Property can be changed.
medical	*false*	Property can be changed.
technical	*false*	Property can be changed.
txPathFault	*false*	Property can be changed.
firedoor	*false*	Property can be changed.
auxMainsFail	*false*	Property can be changed.
auxBattFail	*false*	Property can be changed.
entry-exit-2	*false*	Property can be changed.

value	symbol
0	false
1	true

partSet2

multiplicity: *single (static)*
type: *integer*

Part set 2 property.

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*false*	Property can be changed.
entry-exit	*false*	Property can be changed.
access	*false*	Property can be changed.
fire	*false*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*false*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*false*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*false*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*false*	Property can be changed.
keyswitch	*false*	Property can be changed.
medical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*false*	Property can NOT be changed for this type of zone. Default value must be send.
txPathFault	*false*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*false*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
auxMainsFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*false*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*false*	Property can be changed.

value	symbol
0	false
1	true

ZNM_ACKONKPD

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

ACK on Keypad

The property is only relevant when the property *zoneType* is set to *keyswitch*. Otherwise the value is ignored.

min	max
1	8

ZNM_ACKBYUSR

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

ACK by User

The property is only relevant when the property *zoneType* is set to *keyswitch*. Otherwise the value is ignored.

min	max
1	50

ZNO_TECH_OPTS

multiplicity: *single (static)*
type: *integer*

Tecnical zone reporting option property.

Depending on property *zoneType* the following restrictions are applied:

Zone type	Default value	Restriction
alarm	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
access	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
fire	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
panic	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
24h	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
tamper	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
exitTerminator	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
keyswitch	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
medical	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
technical	*LowTemp*	Property can be changed.
txPathFault	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
firedoor	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.

Zone type	Default value	Restriction
auxMainsFail	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
auxBattFail	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.
entry-exit-2	*LowTemp*	Property can NOT be changed for this type of zone. Default value must be send.

Note: Options `Fire` and `Generic` are available since protocol version `020`.

value	symbol
0	LowTemp
1	HighTemp
2	Gas
3	Water
4	Fire
5	Generic

ZNO_VIRTUAL_ZONE

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Virtual Zone support.

min	max
1	200

value	symbol
0	NONE

HELD_OPEN

multiplicity: *single (static)*
type: *integer*

Enable held open option.

The option is mutually exclusive with *doubleKnock* option.

value	symbol
0	false
1	true

return.Zone

direction: *input*

This is the return message for method "select.Zone".

index

multiplicity: *single (static)*
type: *integer*

Zone index.

min	max
1	128
129	256
257	368

name

multiplicity: *single (static)*
type: *string*

Zone name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

type

multiplicity: *single (static)*
type: *integer*

value	symbol
1	alarm
2	entry-exit
3	access
4	fire
5	panic
6	24h
7	tamper
8	exitTerminator
9	keyswitch
10	medical
11	technical
12	txPathFault
13	firedoor
14	auxMainsFail
15	auxBattFail
16	keybox
17	engReset
18	entry-exit-2

area.1

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 1.

area.2

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 2.

area.3

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 2.

area.4

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 4.

area.5

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 5.

area.6

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 6.

area.7

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 7.

area.8

multiplicity: *single (static)*
type: *boolean*

Relation if zone is assigned to area 8.

partSet

multiplicity: *single (static)*
type: *integer*

Part set.

value	symbol
0	false
1	true

doorBell

multiplicity: *single (static)*
type: *integer*

Door bell.

value	symbol
0	off
1	buzzer
2	siren

dualInput

multiplicity: *single (static)*
type: *integer*

Alarm is generated if two zones in area are active.

value	symbol
0	false
1	true

antiMask

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

soakTest

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

doubleKnock

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

userWalktest

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

engWalktest

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

setKey

multiplicity: *single (static)*
type: *integer*

value	symbol
0	off
1	partSet
2	fullSet
4	partSet2

unSetKey

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

latchKey

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

techPartSet

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

techFullSet

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

techUnset

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

finalDoorSet

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

swingerShunt

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

lcd

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

eee

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csReport

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

log

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

delayT

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

shock

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

inhibit

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

pulseCnt

multiplicity: *single (static)*
type: *integer*

min	max
0	9

grossLvl

multiplicity: *single (static)*
type: *integer*

min	max
0	9

isolate

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

partSet2

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

ZNM_ACKONKPD

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

min	max
1	8

ZNM_ACKBYUSR

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

min	max
1	50

ZNO_TECH_OPTS

multiplicity: *single (static)*
type: *integer*

value	symbol
0	LowTemp
1	HighTemp
2	Gas
3	Water
4	Fire
5	Generic

ZNO_VIRTUAL_ZONE

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Virtual Zone support.

min	max
1	200

value	symbol
0	NONE

HELD_OPEN

multiplicity: *single (static)*
type: *integer*

Enable held open option.

value	symbol
0	false
1	true

return.Area

direction: *input*

This is the incoming message for method "selectArea".

index

multiplicity: *single (static)*
type: *integer*

Area index.

min	max
1	8

name

multiplicity: *single (static)*
type: *string*

Area name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

arEntryTime

multiplicity: *single (static)*
type: *integer*

Area entry time.

min	max
0	255

arExitTime

multiplicity: *single (static)*
type: *integer*

Area exit time.

min	max
0	255

arPreAlarmTime

multiplicity: *single (static)*
type: *integer*

Pre alarm time.

min	max
0	255

arPSnoExitTimer

multiplicity: *single (static)*
type: *integer*

No exit timer.

value	symbol
0	false
1	true

arOptionABmode

multiplicity: *single (static)*
type: *integer*

AB mode.

value	symbol
0	false

value	symbol
1	true

arOptionSirens

multiplicity: *single (static)*
type: *integer*

Sirens.

value	symbol
0	BA
1	BV

arOptionEntryAlarms

multiplicity: *single (static)*
type: *integer*

Area Option: Delayed (0) or instant (1) reporting during entry.

value	symbol
0	Delayed
1	Instant

arOptionAARetry

multiplicity: *single (static)*
type: *integer*

Area Option: Off (0) or On (1).

value	symbol
0	false
1	true

arOptionSilentArm

multiplicity: *single (static)*
type: *integer*

Area Option: Off (0) or On (1).

value	symbol
0	false
1	true

arWarningTime

multiplicity: *single (static)*
type: *integer*

Warning time.

min	max
60	900

arEntryTime2

multiplicity: *single (static)*
type: *integer*

Area entry time.

min	max
0	255

arExitTime2

multiplicity: *single (static)*
type: *integer*

Area exit time.

min	max
0	255

insert.Area

direction: *output*

This is the outgoing message for method "insertArea".

index

multiplicity: *single (static)*
type: *integer*

Area index.

min	max
1	8

name

multiplicity: *single (static)*
type: *string*

Area name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

arEntryTime

multiplicity: *single (static)*
type: *integer*

Area entry time.

min	max
0	255

arExitTime

multiplicity: *single (static)*
type: *integer*

Area exit time.

min	max
0	255

arPreAlarmTime

multiplicity: *single (static)*
type: *integer*

Pre alarm time.

min	max
0	255

arPSnoExitTimer

multiplicity: *single (static)*
type: *integer*

No exit timer.

value	symbol
0	false
1	true

arOptionABmode

multiplicity: *single (static)*
type: *integer*

AB mode.

value	symbol
0	false

value	symbol
1	true

arOptionSirens

multiplicity: *single (static)*
type: *integer*

Sirens.

value	symbol
0	BA
1	BV

arOptionEntryAlarms

multiplicity: *single (static)*
type: *integer*

Area Option: Delayed (0) or instant (1) reporting during entry.

value	symbol
0	Delayed
1	Instant

arOptionAARetry

multiplicity: *single (static)*
type: *integer*

Area Option: Off (0) or On (1).

value	symbol
0	false
1	true

arOptionSilentArm

multiplicity: *single (static)*
type: *integer*

Area Option: Off (0) or On (1).

value	symbol
0	false
1	true

arWarningTime

multiplicity: *single (static)*
type: *integer*

Warning time.

min	max
60	900

arEntryTime2

multiplicity: *single (static)*
type: *integer*

Area entry time.

min	max
0	255

arExitTime2

multiplicity: *single (static)*
type: *integer*

Area exit time.

min	max
0	255

select.Area

direction: *output*

This is the outgoing call for selectArea method.

index

multiplicity: *single (static)*
type: *integer*

Area index.

min	max
1	8

select.User

direction: *output*

This is the outgoing call for selectUser method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

return.User

direction: *input*

This is the return message for method "selectUser" .

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

name

multiplicity: *single (static)*
type: *string*

User name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages

Range	Description
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

userLanguage

multiplicity: *single (static)*
type: *integer*

User language.

value	symbol
208	ENGLISH UK
144	GERMAN
9	DANISH
68	ITALIAN
224	SPANISH
136	FRENCH
64	DUTCH
76	NORWEGIAN-BOKMAL
80	PORTUGUESE
88	SWEDISH
22	POLISH
65	DUTCH BELG
137	FRENCH BELG
25	SLOVAK
13	FINNISH
8	CZECH
28	TURKISH
6	CATALAN
16	HUNGARIAN

userGroup1

multiplicity: *single (static)*
type: *integer*

User group 1.

min	max
0	16

userGroup2

multiplicity: *single (static)*
type: *integer*

User group 2.

min	max
0	16

userGroup3

multiplicity: *single (static)*
type: *integer*

User group 3.

min	max
0	16

userGroup4

multiplicity: *single (static)*
type: *integer*

User group 4.

min	max
0	16

userGroup5

multiplicity: *single (static)*
type: *integer*

User group 5.

min	max
0	16

userGroup6

multiplicity: *single (static)*
type: *integer*

User group 6.

min	max
0	16

userGroup7

multiplicity: *single (static)*
type: *integer*

User group 7.

min	max
0	16

userGroup8

multiplicity: *single (static)*
type: *integer*

User group 8.

min	max
0	16

userPhoneNum

multiplicity: *single (static)*
type: *string*

Phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters:: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

insert.User

direction: *output*

This is the outgoing message for method "insertUser" .

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

name

multiplicity: *single (static)*
type: *string*

User name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages

Range	Description
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

userLanguage

multiplicity: *single (static)*
type: *integer*

User language.

value	symbol
208	ENGLISH UK
144	GERMAN
9	DANISH
68	ITALIAN
224	SPANISH
136	FRENCH
64	DUTCH
76	NORWEGIAN-BOKMAL
80	PORTUGUESE
88	SWEDISH
22	POLISH
65	DUTCH BELG
137	FRENCH BELG
25	SLOVAK
13	FINNISH
8	CZECH
28	TURKISH
6	CATALAN
16	HUNGARIAN

userGroup1

multiplicity: *single (static)*
type: *integer*

User group 1.

min	max
0	16

userGroup2

multiplicity: *single (static)*
type: *integer*

User group 2.

min	max
0	16

userGroup3

multiplicity: *single (static)*
type: *integer*

User group 3.

min	max
0	16

userGroup4

multiplicity: *single (static)*
type: *integer*

User group 4.

min	max
0	16

userGroup5

multiplicity: *single (static)*
type: *integer*

User group 5.

min	max
0	16

userGroup6

multiplicity: *single (static)*
type: *integer*

User group 6.

min	max
0	16

userGroup7

multiplicity: *single (static)*
type: *integer*

User group 7.

min	max
0	16

userGroup8

multiplicity: *single (static)*
type: *integer*

User group 8.

min	max
0	16

userPhoneNum

multiplicity: *single (static)*
type: *string*

Phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters:: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

add.Users

direction: *output*

This is the outgoing message for method "addUsers". Use this method for sync.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

name

multiplicity: *single (static)*
type: *string*

User name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages

Range	Description
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

userLanguage

multiplicity: *single (static)*
type: *integer*

User language.

value	symbol
208	ENGLISH UK
144	GERMAN
9	DANISH
68	ITALIAN
224	SPANISH
136	FRENCH
64	DUTCH
76	NORWEGIAN-BOKMAL
80	PORTUGUESE
88	SWEDISH
22	POLISH
65	DUTCH BELG
137	FRENCH BELG
25	SLOVAK
13	FINNISH
8	CZECH
28	TURKISH
6	CATALAN
16	HUNGARIAN

userGroup1

multiplicity: *single (static)*
type: *integer*

User group 1.

min	max
0	16

userGroup2

multiplicity: *single (static)*
type: *integer*

User group 2.

min	max
0	16

userGroup3

multiplicity: *single (static)*
type: *integer*

User group 3.

min	max
0	16

userGroup4

multiplicity: *single (static)*
type: *integer*

User group 4.

min	max
0	16

userGroup5

multiplicity: *single (static)*
type: *integer*

User group 5.

min	max
0	16

userGroup6

multiplicity: *single (static)*
type: *integer*

User group 6.

min	max
0	16

userGroup7

multiplicity: *single (static)*
type: *integer*

User group 7.

min	max
0	16

userGroup8

multiplicity: *single (static)*
type: *integer*

User group 8.

min	max
0	16

userPIN

multiplicity: *single (static)*
type: *string*

User PIN code.

userCARD

multiplicity: *single (static)*
type: *string*

User card data.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

userPhoneNum

multiplicity: *single (static)*
type: *string*

Phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters:: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

add.RemoteUsers

direction: *output*

This is the outgoing message for method "addRemoteUsers". Use this method for sync.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userPIN

multiplicity: *single (static)*
type: *string*

User remote PIN code.

select.CS

direction: *output*

This is the outgoing call for selectCS method.

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

selectV.CS

direction: *output*

This is the outgoing call for selectCS method, variant part.

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

return.CS

direction: *input*

This is the return message for method "selectCS"

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

CS name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

phoneNum

multiplicity: *single (static)*
type: *string*

Phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters:: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

cs_PROT

multiplicity: *single (static)*
type: *integer*

CS protocol.

value	symbol
1	CID
2	SIA
3	XSIA
4	VOICE
9	OHCID
10	OHSIA
11	OHXSIA
13	SMS
14	SMS+CID
15	SMS+SIA
16	SMS+XSIA
17	SMS+VOICE
19	STELSIA
20	STELXSIA
21	CSIA
22	CVDS
23	OHVIDEO+CID
24	OHVIDEO+SIA
25	OHVIDEO+XSIA
27	ATS75XX+XSIA
28	PHOTO+XSIA
31	SMS+MMS

cs_DIALER

multiplicity: *single (static)*
type: *integer*

CS dialer.

min	max
1	7

csEVPERCALL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csBACKUP

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csACCPERCALL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIAAREAMODIF

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIA32NAMECHR

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csEventNumber

multiplicity: *single (static)*
type: *integer*

00 - 4 digits 01 - 3 digits 10 - 2 digits

value	symbol
0	4-digits
1	3-digits
2	2-digits

csFREQBELL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csAlwaysCall

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_FTC

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_NO_ACK

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

cs_Retrycnt

multiplicity: *single (static)*
type: *integer*

min	max
0	14

cs_SUBEV_CODING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

CSNM_OPCL_REPORTING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

return.CS_2

direction: *input*

This is the return message for method "selectCS"

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

CS name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

cs_PROT

multiplicity: *single (static)*
type: *integer*

CS protocol.

value	symbol
1	CID
2	SIA
3	XSIA
4	VOICE
9	OHCID
10	OHSIA
11	OHXSIA
13	SMS
14	SMS+CID
15	SMS+SIA
16	SMS+XSIA
17	SMS+VOICE
19	STELSIA
20	STELXSIA
21	CSIA
22	CVDS
23	OHVIDEO+CID
24	OHVIDEO+SIA
25	OHVIDEO+XSIA
27	ATS75XX+XSIA
28	PHOTO+XSIA
31	SMS+MMS

cs_DIALER

multiplicity: *single (static)*
type: *integer*

CS dialer.

min	max
1	7

csEVPERCALL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csBACKUP

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csACCPERCALL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

CSNM_OPCL_REPORTING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csAlwaysCall

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

cs_SUBEV_CODING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIAAREAMODIF

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIA32NAMECHR

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csEventNumber

multiplicity: *single (static)*
type: *integer*

00 - 4 digits 01 - 3 digits 10 - 2 digits

value	symbol
0	4-digits
1	3-digits
2	2-digits

csFREQBELL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_FTC

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_NO_ACK

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

cs_Retrycnt

multiplicity: *single (static)*
type: *integer*

min	max
0	14

phoneNum

multiplicity: *single (static)*
type: *string*

Phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters:: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

CSNM_IP_ADDRESS

multiplicity: *single (static)*
type: *string*

IP Address of central station.

CSNM_IP_PORT

multiplicity: *single (static)*
type: *integer*

CS IP port

CSNM_OHIP_VERSION

multiplicity: *single (static)*
type: *integer*

OH version

value	symbol
0	OHVER_STD
1	OHVER_ENH
2	OHVER_GRADE4

CSNM_HBPERIOD

multiplicity: *single (static)*
type: *integer*

Heartbeat period [s]

min	max
1	86399

return.CS_CMN

direction: *input*

This is the return message for method "selectCS"

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

CS name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

cs_PROT

multiplicity: *single (static)*
type: *integer*

CS protocol.

value	symbol
1	CID
2	SIA
3	XSIA
4	VOICE
9	OHCID
10	OHSIA
11	OHXSIA
13	SMS
14	SMS+CID
15	SMS+SIA
16	SMS+XSIA
17	SMS+VOICE
19	STELSIA
20	STELXSIA
21	CSIA
22	CVDS
23	OHVIDEO+CID
24	OHVIDEO+SIA
25	OHVIDEO+XSIA
27	ATS75XX+XSIA
28	PHOTO+XSIA
31	SMS+MMS

cs_DIALER

multiplicity: *single (static)*
type: *integer*

CS dialer.

min	max
1	7

csBACKUP

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csACCPERCALL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

CSNM_OPCL_REPORTING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

cs_SUBEV_CODING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIAAREAMODIF

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIA32NAMECHR

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csEventNumber

multiplicity: *single (static)*
type: *integer*

00 - 4 digits 01 - 3 digits 10 - 2 digits

value	symbol
0	4-digits
1	3-digits
2	2-digits

csFREQBELL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_FTC

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_NO_ACK

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

cs_Retrycnt

multiplicity: *single (static)*
type: *integer*

min	max
0	14

CSNM_OHIP_VERSION

multiplicity: *single (static)*
type: *integer*

OH version

value	symbol
0	OHVER_STD
1	OHVER_ENH
2	OHVER_GRADE4

CSNM_HBPERIOD

multiplicity: *single (static)*
type: *integer*

Heartbeat period [s]

min	max
1	86399

csREPORT_TYPE

multiplicity: *single (static)*
type: *integer*

Reporting type

value	symbol
0	CS_REP_PHONE
1	CS_REP_USER
2	CS_REP_UG

csRP_TIME

multiplicity: *single (static)*
type: *integer*

Auto test call time

min	max
0	1439

csRP_PERIOD

multiplicity: *single (static)*
type: *integer*

Auto test call period

min	max
0	999

csRP_EXTEND

multiplicity: *single (static)*
type: *integer*

Automatic Test Call Option: Extend test call

value	symbol
0	false
1	true

csRP_FREQFTC

multiplicity: *single (static)*
type: *integer*

Automatic Test Call Option: Frequent test calls in case of FTC

value	symbol
0	false
1	true

CSNM_OHIP_RECEIVER

multiplicity: *single (static)*
type: *string*

OH receiver number

CSNM_OHIP_LINE

multiplicity: *single (static)*
type: *string*

OH line number

csRP_VIDEO_ENCRYPTION

multiplicity: *single (static)*
type: *integer*

Video encryption

value	symbol
0	false
1	true

csRP_IP_VIDEO_PORT

multiplicity: *single (static)*
type: *integer*

CS IP port

return.CS_PHONE

direction: *input*

This is the return message for method "selectCS",variant part with PHONE num

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

phoneNum

multiplicity: *single (static)*
type: *string*

Phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters:: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

return.CS_IP

direction: *input*

This is the return message for method "selectCS",variant part with IP addr

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

CSNM_IP_ADDRESS

multiplicity: *single (static)*
type: *string*

IP Address of central station.

CSNM_IP_PORT

multiplicity: *single (static)*
type: *integer*

CS IP port

return.CS_USER

direction: *input*

This is the return message for method "selectCS",variant part with User ID

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

CSNM_USER

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

User ID.

min	max
1	50

return.CS_USERGROUP

direction: *input*

This is the return message for method "selectCS",variant part with User Group

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

CSNM_USERGROUP

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

User ID.

min	max
1	16

insert.CS

direction: *output*

This is the outgoing message for method "insertCS"

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

CS name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

phoneNum

multiplicity: *single (static)*
type: *string*

phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters:: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

cs_PROT

multiplicity: *single (static)*
type: *integer*

CS protocol.

value	symbol
1	CID
2	SIA
3	XSIA
4	VOICE
9	OHCID
10	OHSIA
11	OHXSIA
13	SMS
14	SMS+CID
15	SMS+SIA
16	SMS+XSIA
17	SMS+VOICE
19	STELSIA
20	STELXSIA
21	CSIA
22	CVDS
23	OHVIDEO+CID
24	OHVIDEO+SIA
25	OHVIDEO+XSIA
27	ATS75XX+XSIA
28	PHOTO+XSIA
31	SMS+MMS

cs_DIALER

multiplicity: *single (static)*
type: *integer*

CS dialer.

min	max
1	7

csEVPERCALL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csBACKUP

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csACCPERCALL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIAAREAMODIF

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIA32NAMECHR

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csEventNumber

multiplicity: *single (static)*
type: *integer*

00 - 4 digits 01 - 3 digits 10 - 2 digits

value	symbol
0	4-digits
1	3-digits
2	2-digits

csFREQBELL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csAlwaysCall

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_FTC

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_NO_ACK

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

cs_Retrycnt

multiplicity: *single (static)*
type: *integer*

min	max
0	14

cs_SUBEV_CODING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

CSNM_OPCL_REPORTING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

insert.CS_2

direction: *output*

This is the outgoing message for method "insertCS"

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

CS name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

cs_PROT

multiplicity: *single (static)*
type: *integer*

CS protocol.

value	symbol
1	CID
2	SIA
3	XSIA
4	VOICE
9	OHCID
10	OHSIA
11	OHXSIA
13	SMS
14	SMS+CID
15	SMS+SIA
16	SMS+XSIA
17	SMS+VOICE
19	STELSIA
20	STELXSIA
21	CSIA
22	CVDS
23	OHVIDEO+CID
24	OHVIDEO+SIA
25	OHVIDEO+XSIA
27	ATS75XX+XSIA
28	PHOTO+XSIA
31	SMS+MMS

cs_DIALER

multiplicity: *single (static)*
type: *integer*

CS dialer.

min	max
1	7

csEVPERCALL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csBACKUP

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csACCPERCALL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

CSNM_OPCL_REPORTING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csAlwaysCall

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

cs_SUBEV_CODING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIAAREAMODIF

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIA32NAMECHR

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csEventNumber

multiplicity: *single (static)*
type: *integer*

00 - 4 digits 01 - 3 digits 10 - 2 digits

value	symbol
0	4-digits
1	3-digits
2	2-digits

csFREQBELL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_FTC

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_NO_ACK

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

cs_Retrycnt

multiplicity: *single (static)*
type: *integer*

min	max
0	14

phoneNum

multiplicity: *single (static)*
type: *string*

Phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters:: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

CSNM_IP_ADDRESS

multiplicity: *single (static)*
type: *string*

IP Address of central station.

CSNM_IP_PORT

multiplicity: *single (static)*
type: *integer*

CS IP port

CSNM_OHIP_VERSION

multiplicity: *single (static)*
type: *integer*

OH version

value	symbol
0	OHVER_STD
1	OHVER_ENH
2	OHVER_GRADE4

CSNM_HBPERIOD

multiplicity: *single (static)*
type: *integer*

Heartbeat period [s]

min	max
1	86399

insert.CS_CMN

direction: *output*

This is the outgoing message for method "insertCS"

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

CS name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

cs_PROT

multiplicity: *single (static)*
type: *integer*

CS protocol.

value	symbol
1	CID
2	SIA
3	XSIA
4	VOICE
9	OHCID
10	OHSIA
11	OHXSIA
13	SMS
14	SMS+CID
15	SMS+SIA
16	SMS+XSIA
17	SMS+VOICE
19	STELSIA
20	STELXSIA
21	CSIA
22	CVDS
23	OHVIDEO+CID
24	OHVIDEO+SIA
25	OHVIDEO+XSIA
27	ATS75XX+XSIA
28	PHOTO+XSIA
31	SMS+MMS

cs_DIALER

multiplicity: *single (static)*
type: *integer*

CS dialer.

min	max
1	7

csBACKUP

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csACCPERCALL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

CSNM_OPCL_REPORTING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csAlwaysCall

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

cs_SUBEV_CODING

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIAAREAMODIF

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csSIA32NAMECHR

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csEventNumber

multiplicity: *single (static)*
type: *integer*

00 - 4 digits 01 - 3 digits 10 - 2 digits

value	symbol
0	4-digits
1	3-digits
2	2-digits

csFREQBELL

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_FTC

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

csVOICE_NO_ACK

multiplicity: *single (static)*
type: *integer*

value	symbol
0	false
1	true

cs_Retrycnt

multiplicity: *single (static)*
type: *integer*

min	max
0	14

CSNM_OHIP_VERSION

multiplicity: *single (static)*
type: *integer*

OH version

value	symbol
0	OHVER_STD
1	OHVER_ENH
2	OHVER_GRADE4

CSNM_HBPERIOD

multiplicity: *single (static)*
type: *integer*

Heartbeat period [s]

min	max
1	86399

csREPORT_TYPE

multiplicity: *single (static)*
type: *integer*

Reporting type

value	symbol
0	CS_REP_PHONE
1	CS_REP_USER
2	CS_REP_UG

csRP_TIME

multiplicity: *single (static)*
type: *integer*

Auto test call time

min	max
0	1439

csRP_PERIOD

multiplicity: *single (static)*
type: *integer*

Auto test call period

min	max
0	999

csRP_EXTEND

multiplicity: *single (static)*
type: *integer*

Automatic Test Call Option: Extend test call

value	symbol
0	false
1	true

csRP_FREQFTC

multiplicity: *single (static)*
type: *integer*

Automatic Test Call Option: Frequent test calls in case of FTC

value	symbol
0	false
1	true

CSNM_OHIP_RECEIVER

multiplicity: *single (static)*
type: *string*

OH receiver number

CSNM_OHIP_LINE

multiplicity: *single (static)*
type: *string*

OH line number

csRP_VIDEO_ENCRYPTION

multiplicity: *single (static)*
type: *integer*

Video encryption

value	symbol
0	false
1	true

csRP_IP_VIDEO_PORT

multiplicity: *single (static)*
type: *integer*

Video port

insertV.CS_PHONE

direction: *output*

This is the outgoing message for method "insertCS", variant part with PHONE num.

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

phoneNum

multiplicity: *single (static)*
type: *string*

Phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters:: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

insertV.CS_IP

direction: *output*

This is the outgoing message for method "insertCS", variant part with IP addr.

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

CSNM_IP_ADDRESS

multiplicity: *single (static)*
type: *string*

IP Address of central station.

CSNM_IP_PORT

multiplicity: *single (static)*
type: *integer*

CS IP port

insertV.CS_USER

direction: *output*

This is the insert message for method "insertCS",variant part with User ID

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

CSNM_USER

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

User ID.

min	max
1	50

insertV.CS_USERGROUP

direction: *output*

This is the insert message for method "insertCS",variant part with User Group

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

CSNM_USERGROUP

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

User ID.

min	max
1	16

select.RAS

direction: *output*

This is the outgoing call for "selectRAS" method.

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

selectV.RASAct

direction: *output*

This is the outgoing call for "selectVRASAct" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

return.RAS

direction: *input*

This is the return message for "selectRAS" method.

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

name

multiplicity: *single (static)*
type: *string*

RAS name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

rasTampArea

multiplicity: *single (static)*
type: *integer*

rasTampArea.

min	max
1	8

rasViewArea.1

multiplicity: *single (static)*
type: *boolean*

rasViewArea 1.

rasViewArea.2

multiplicity: *single (static)*
type: *boolean*

rasViewArea 2.

rasViewArea.3

multiplicity: *single (static)*
type: *boolean*

rasViewArea 3.

rasViewArea.4

multiplicity: *single (static)*
type: *boolean*

rasViewArea 4.

rasViewArea.5

multiplicity: *single (static)*
type: *boolean*

rasViewArea 5.

rasViewArea.6

multiplicity: *single (static)*
type: *boolean*

rasViewArea 6.

rasViewArea.7

multiplicity: *single (static)*
type: *boolean*

rasViewArea 7.

rasViewArea.8

multiplicity: *single (static)*
type: *boolean*

rasViewArea 8.

rasCtrlArea.1

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 1.

rasCtrlArea.2

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 2.

rasCtrlArea.3

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 3.

rasCtrlArea.4

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 4.

rasCtrlArea.5

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 5.

rasCtrlArea.6

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 6.

rasCtrlArea.7

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 7.

rasCtrlArea.8

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 8.

ras3XBADGE

multiplicity: *single (static)*
type: *integer*

3 x time badge settings.

Note: For protocol version below `020` the only valid options are *Off* and *On*.

value	symbol
0	Off
1	On
2	PSet1
3	PSet2

rasCARDPIN

multiplicity: *single (static)*
type: *integer*

Card and PIN settings.

value	symbol
0	Card or PIN
1	PIN only
2	Card only
3	Card and PIN unset
4	Card and PIN always

rasCardUnset

multiplicity: *single (static)*
type: *integer*

Card unset.

Note: For protocol version below 020 the only valid options are *Off*, *Unset* and *SetUnset*.

value	symbol
0	Off
1	Unset
2	SetUnset
3	PSet1Unset
4	PSet2Unset

RASM_EEPINLOCK

multiplicity: *single (static)*
type: *integer*

RASM_EEPINLOCK.

value	symbol
0	false
1	true

RASM_ISOLATE

multiplicity: *single (static)*
type: *integer*

RASM_ISOLATE.

value	symbol
0	false
1	true

RASM_BUZSIL

multiplicity: *single (static)*
type: *integer*

Buzzer silent mode.

value	symbol
0	Never
1	During partset

value	symbol
2	Always
3	During E/E

RASM_QSET

multiplicity: *single (static)*
type: *integer*

QuickSet mode.

value	symbol
0	Off
1	Basic
2	Advanced

rasDoorAcces

multiplicity: *single (static)*
type: *integer*

Door access.

value	symbol
0	false
1	true

RASM_AILED1.1

multiplicity: *single (static)*
type: *boolean*

Map Area 1 to LED1.

RASM_AILED1.2

multiplicity: *single (static)*
type: *boolean*

Map Area 2 to LED1.

RASM_AILED1.3

multiplicity: *single (static)*
type: *boolean*

Map Area 3 to LED1.

RASM_AILED1.4

multiplicity: *single (static)*
type: *boolean*

Map Area 4 to LED1.

RASM_AILED1.5

multiplicity: *single (static)*
type: *boolean*

Map Area 5 to LED1.

RASM_AILED1.6

multiplicity: *single (static)*
type: *boolean*

Map Area 6 to LED1.

RASM_AILED1.7

multiplicity: *single (static)*
type: *boolean*

Map Area 7 to LED1.

RASM_AILED1.8

multiplicity: *single (static)*
type: *boolean*

Map Area 8 to LED1.

RASM_AILED1

multiplicity: *single (static)*
type: *integer*

Area indication for LED 1.

RASM_AILED2.1

multiplicity: *single (static)*
type: *boolean*

Map Area 1 to LED2.

RASM_AILED2.2

multiplicity: *single (static)*
type: *boolean*

Map Area 2 to LED2.

RASM_AILED2.3

multiplicity: *single (static)*
type: *boolean*

Map Area 3 to LED2.

RASM_AILED2.4

multiplicity: *single (static)*
type: *boolean*

Map Area 4 to LED2.

RASM_AILED2.5

multiplicity: *single (static)*
type: *boolean*

Map Area 5 to LED2.

RASM_AILED2.6

multiplicity: *single (static)*
type: *boolean*

Map Area 6 to LED2.

RASM_AILED2.7

multiplicity: *single (static)*
type: *boolean*

Map Area 7 to LED2.

RASM_AILED2.8

multiplicity: *single (static)*
type: *boolean*

Map Area 8 to LED2.

RASM_AILED2

multiplicity: *single (static)*
type: *integer*

Area indication for LED 1.

RASM_AREAINTERFACE

multiplicity: *single (static)*
type: *integer*

Area interface type.

value	symbol
0	List
1	Symbolic

RASM_LCDBACKLIGHT

multiplicity: *single (static)*
type: *integer*

LCD backlight mode.

value	symbol
0	Normal
1	AlwaysOn
2	ExclEntry

RASM_ACKONKPD

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

ACK on Keypad.

min	max
1	8

RASM_EE1BUZ

multiplicity: *single (static)*
type: *integer*

RAS buzzer during EE1.

value	symbol
0	false

value	symbol
1	true

RASM_EE2BUZ

multiplicity: *single (static)*
type: *integer*

RAS buzzer during EE2.

value	symbol
0	false
1	true

RASM_PANICQKEY

multiplicity: *single (static)*
type: *integer*

RAS panic quick key enable.

value	symbol
0	false
1	true

insert.RAS

direction: *output*

This is the outgoing message for "insertRAS" method.

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

name

multiplicity: *single (static)*
type: *string*

RAS name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

rasTampArea

multiplicity: *single (static)*
type: *integer*

rasTampArea.

min	max
1	8

rasViewArea.1

multiplicity: *single (static)*
type: *boolean*

rasViewArea 1.

rasViewArea.2

multiplicity: *single (static)*
type: *boolean*

rasViewArea 2.

rasViewArea.3

multiplicity: *single (static)*
type: *boolean*

rasViewArea 3.

rasViewArea.4

multiplicity: *single (static)*
type: *boolean*

rasViewArea 4.

rasViewArea.5

multiplicity: *single (static)*
type: *boolean*

rasViewArea 5.

rasViewArea.6

multiplicity: *single (static)*
type: *boolean*

rasViewArea 6.

rasViewArea.7

multiplicity: *single (static)*
type: *boolean*

rasViewArea.8

multiplicity: *single (static)*
type: *boolean*

rasViewArea 8.

rasCtrlArea.1

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 1.

rasCtrlArea.2

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 2.

rasCtrlArea.3

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 3.

rasCtrlArea.4

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 4.

rasCtrlArea.5

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 5.

rasCtrlArea.6

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 6.

rasCtrlArea.7

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 7.

rasCtrlArea.8

multiplicity: *single (static)*
type: *boolean*

rasCtrlArea 8.

ras3XBADGE

multiplicity: *single (static)*
type: *integer*

3 x time badge settings.

Note: For protocol version below 020 the only valid options are *Off* and *On*.

value	symbol
0	Off
1	On
2	PSet1
3	PSet2

rasCARDPIN

multiplicity: *single (static)*
type: *integer*

Card and PIN settings.

value	symbol
0	Card or PIN
1	PIN only
2	Card only
3	Card and PIN unset
4	Card and PIN always

rasCardUnset

multiplicity: *single (static)*
type: *integer*

Card unset / one time badge.

Note: For protocol version below 020 the only valid options are *Off*, *Unset* and *SetUnset*.

value	symbol
0	Off
1	Unset
2	SetUnset
3	PSet1Unset
4	PSet2Unset

RASM_EEPINLOCK

multiplicity: *single (static)*
type: *integer*

RASM_EEPINLOCK.

value	symbol
0	false
1	true

RASM_ISOLATE

multiplicity: *single (static)*
type: *integer*

RASM_ISOLATE.

value	symbol
0	false
1	true

RASM_BUZSIL

multiplicity: *single (static)*
type: *integer*

Buzzer silent mode.

value	symbol
0	Never
1	During partset

value	symbol
2	Always
3	During E/E

RASM_QSET

multiplicity: *single (static)*
type: *integer*

QuickSet mode.

value	symbol
0	Off
1	Basic
2	Advanced

rasDoorAcces

multiplicity: *single (static)*
type: *integer*

Door access.

value	symbol
0	false
1	true

RASM_AILED1.1

multiplicity: *single (static)*
type: *boolean*

Map Area 1 to LED1.

RASM_AILED1.2

multiplicity: *single (static)*
type: *boolean*

Map Area 2 to LED1.

RASM_AILED1.3

multiplicity: *single (static)*
type: *boolean*

Map Area 3 to LED1.

RASM_AILED1.4

multiplicity: *single (static)*
type: *boolean*

Map Area 4 to LED1.

RASM_AILED1.5

multiplicity: *single (static)*
type: *boolean*

Map Area 5 to LED1.

RASM_AILED1.6

multiplicity: *single (static)*
type: *boolean*

Map Area 6 to LED1.

RASM_AILED1.7

multiplicity: *single (static)*
type: *boolean*

Map Area 7 to LED1.

RASM_AILED1.8

multiplicity: *single (static)*
type: *boolean*

Map Area 8 to LED1.

RASM_AILED1

multiplicity: *single (static)*
type: *integer*

Area indication for LED 1.

RASM_AILED2.1

multiplicity: *single (static)*
type: *boolean*

Map Area 1 to LED2.

RASM_AILED2.2

multiplicity: *single (static)*
type: *boolean*

Map Area 2 to LED2.

RASM_AILED2.3

multiplicity: *single (static)*
type: *boolean*

Map Area 3 to LED2.

RASM_AILED2.4

multiplicity: *single (static)*
type: *boolean*

Map Area 4 to LED2.

RASM_AILED2.5

multiplicity: *single (static)*
type: *boolean*

Map Area 5 to LED2.

RASM_AILED2.6

multiplicity: *single (static)*
type: *boolean*

Map Area 6 to LED2.

RASM_AILED2.7

multiplicity: *single (static)*
type: *boolean*

Map Area 7 to LED2.

RASM_AILED2.8

multiplicity: *single (static)*
type: *boolean*

Map Area 8 to LED2.

RASM_AILED2

multiplicity: *single (static)*
type: *integer*

Area indication for LED 1.

RASM_AREAINTERFACE

multiplicity: *single (static)*
type: *integer*

Area interface type.

value	symbol
0	List
1	Symbolic

RASM_LCDBACKLIGHT

multiplicity: *single (static)*
type: *integer*

LCD backlight mode.

value	symbol
0	Normal
1	AlwaysOn
2	ExclEntry

RASM_ACKONKPD

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

ACK on Keypad.

min	max
1	8

RASM_EE1BUZ

multiplicity: *single (static)*
type: *integer*

RAS buzzer during EE1.

value	symbol
0	false

value	symbol
1	true

RASM_EE2BUZ

multiplicity: *single (static)*
type: *integer*

RAS buzzer during EE2.

value	symbol
0	false
1	true

RASM_PANICQKEY

multiplicity: *single (static)*
type: *integer*

RAS panic quick key enable.

value	symbol
0	false
1	true

insertV.RASActNone

direction: *output*

This is the outgoing call for "insertVRASActNone" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActSet

direction: *output*

This is the outgoing call for "insertVRASActSet" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

SETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Set Area 1.

SETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Set Area 2.

SETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Set Area 3.

SETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Set Area 4.

SETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Set Area 5.

SETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Set Area 6.

SETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Set Area 7.

SETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Set Area 8.

SETM_AREAS

multiplicity: *single (static)*
type: *integer*

Set Areas.

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActUnset

direction: *output*

This is the outgoing call for "insertVRASActUnset" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

UNSETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Unset Area 1.

UNSETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Unset Area 2.

UNSETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Unset Area 3.

UNSETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Unset Area 4.

UNSETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Unset Area 5.

UNSETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Unset Area 6.

UNSETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Unset Area 7.

UNSETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Unset Area 8.

UNSETM_AREAS

multiplicity: *single (static)*
type: *integer*

Unset Areas.

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActTrigger

direction: *output*

This is the outgoing call for "insertVRASActTrigger" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

TRIGGERM_INDEX

multiplicity: *single (static)*
type: *integer*

Trigger index.

min	max
1	255

TRIGGERM_STATE

multiplicity: *single (static)*
type: *integer*

Trigger state.

value	symbol
0	CLEAR
1	SET
2	TOGGLE

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActDoorbell

direction: *output*

This is the outgoing call for "insertVRASActDoorbell" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

DOORBELLM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 1.

DOORBELLM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 2.

DOORBELLM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 3.

DOORBELLM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 4.

DOORBELLM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 5.

DOORBELLM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 6.

DOORBELLM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 7.

DOORBELLM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 8.

DOORBELLM_AREAS

multiplicity: *single (static)*
type: *integer*

Doorbell Areas.

DOORBELLM_STATE

multiplicity: *single (static)*
type: *integer*

Doorbell state.

value	symbol
0	CLEAR
1	SET
2	TOGGLE

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActPSet1

direction: *output*

This is the outgoing call for "insertVRASActPSet1" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

PSET1M_AREAS.1

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 1.

PSET1M_AREAS.2

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 2.

PSET1M_AREAS.3

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 3.

PSET1M_AREAS.4

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 4.

PSET1M_AREAS.5

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 5.

PSET1M_AREAS.6

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 6.

PSET1M_AREAS.7

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 7.

PSET1M_AREAS.8

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 8.

PSET1M_AREAS

multiplicity: *single (static)*
type: *integer*

PartSet1 Areas.

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActPSet2

direction: *output*

This is the outgoing call for "insertVRASActPSet2" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

PSET2M_AREAS.1

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 1.

PSET2M_AREAS.2

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 2.

PSET2M_AREAS.3

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 3.

PSET2M_AREAS.4

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 4.

PSET2M_AREAS.5

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 5.

PSET2M_AREAS.6

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 6.

PSET2M_AREAS.7

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 7.

PSET2M_AREAS.8

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 8.

PSET2M_AREAS

multiplicity: *single (static)*
type: *integer*

PartSet2 Areas.

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActInh

direction: *output*

This is the outgoing call for "insertVRASActInh" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActTCall

direction: *output*

This is the outgoing call for "insertVRASActTCall" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActPCC

direction: *output*

This is the outgoing call for "insertVRASActPCC" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActServIn

direction: *output*

This is the outgoing call for "insertVRASActServIn" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActPanic

direction: *output*

This is the outgoing call for "insertVRASActPanic" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActDoorbellRAS

direction: *output*

This is the outgoing call for "insertVRASActDoorbellRAS" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

DOORBELLM_RAS.1

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 1.

DOORBELLM_RAS.2

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 2.

DOORBELLM_RAS.3

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 3.

DOORBELLM_RAS.4

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 4.

DOORBELLM_RAS.5

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 5.

DOORBELLM_RAS.6

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 6.

DOORBELLM_RAS.7

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 7.

DOORBELLM_RAS.8

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 8.

DOORBELLM_RAS

multiplicity: *single (static)*
type: *integer*

Doorbell Areas.

DOORBELLRASM_STATE

multiplicity: *single (static)*
type: *integer*

Doorbell state.

value	symbol
0	CLEAR
1	SET
2	TOGGLE

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActSetWET

direction: *output*

This is the outgoing call for "insertVRASActSetWET" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

SETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Set Area 1.

SETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Set Area 2.

SETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Set Area 3.

SETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Set Area 4.

SETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Set Area 5.

SETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Set Area 6.

SETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Set Area 7.

SETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Set Area 8.

SETM_AREAS

multiplicity: *single (static)*
type: *integer*

Set Areas.

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActFireReset

direction: *output*

This is the outgoing call for "insertVRASActFireReset" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

SETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Set Area 1.

SETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Set Area 2.

SETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Set Area 3.

SETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Set Area 4.

SETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Set Area 5.

SETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Set Area 6.

SETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Set Area 7.

SETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Set Area 8.

SETM_AREAS

multiplicity: *single (static)*
type: *integer*

Set Areas.

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActOpenZn

direction: *output*

This is the outgoing call for "insertVRASOpenZn" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActAlarmZn

direction: *output*

This is the outgoing call for "insertVRASAlarmZn" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActFaults

direction: *output*

This is the outgoing call for "insertVRASFaults" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActAlarmMem

direction: *output*

This is the outgoing call for "insertVRASAlarmMem" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActZonesAck

direction: *output*

This is the outgoing call for "insertVRASZonesAck" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActWalkTest

direction: *output*

This is the outgoing call for "insertVRASActWalkTest" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

WALKTESTM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 1.

WALKTESTM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 2.

WALKTESTM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 3.

WALKTESTM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 4.

WALKTESTM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 5.

WALKTESTM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 6.

WALKTESTM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 7.

WALKTESTM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 8.

WALKTESTM_AREAS

multiplicity: *single (static)*
type: *integer*

Walk Test Areas.

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActOutputTest

direction: *output*

This is the outgoing call for "insertVRASActOutputTest" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

OUTPUTTESTM_OUTPUT.1

multiplicity: *single (static)*
type: *integer*

Output test 1.

min	max
1	200

value	symbol
0	NONE

OUTPUTTESTM_OUTPUT.2

multiplicity: *single (static)*
type: *integer*

Output test 2.

min	max
1	200

value	symbol
0	NONE

OUTPUTTESTM_OUTPUT.3

multiplicity: *single (static)*
type: *integer*

Output test 3.

min	max
1	200

value	symbol
0	NONE

OUTPUTTESTM_OUTPUT.4

multiplicity: *single (static)*
type: *integer*

Output test 4.

min	max
1	200

value	symbol
0	NONE

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActFire

direction: *output*

This is the outgoing call for "insertVRASActFire" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActMedical

direction: *output*

This is the outgoing call for "insertVRASActMedical" method.

subindex

multiplicity: *single (static)*
type: *integer*

RAS function key index.

min	max
1	7

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

insertV.RASActTakePicture

direction: *output*

This is the outgoing call for "insertVFOBTakePicture" method.

subindex

multiplicity: *single (static)*
type: *integer*

FOB button index.

min	max
1	10

CAMERA_ID

multiplicity: *single (static)*
type: *integer*

Camera ID.

min	max
1	128
257	368

CS_NUMBER

multiplicity: *single (static)*
type: *integer*

Central station number.

min	max
1	16

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

select.FOB

direction: *output*

This is the outgoing call for "selectFOB" method.

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

selectV.FobAct

direction: *output*

This is the outgoing call for "selectVFobAct" method.

subindex

multiplicity: *single (static)*
type: *integer*

FOB button index.

min	max
1	10

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

return.FOB

direction: *input*

This is the return message for "selectFOB" method.

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

name

multiplicity: *single (static)*
type: *string*

FOB name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

fobUserID

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

fobUserID.

min	max
1	50

insert.FOB

direction: *output*

This is the outgoing message for "insertFOB" method.

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

name

multiplicity: *single (static)*
type: *string*

FOB name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

fobUserID

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

fobUserID.

min	max
1	50

insertV.FOBActNone

direction: *output*

This is the outgoing call for "insertVFOBActNone" method.

subindex

multiplicity: *single (static)*
type: *integer*

FOB button index.

min	max
1	10

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

insertV.FOBActSet

direction: *output*

This is the outgoing call for "insertVFOBActSet" method.

subindex

multiplicity: *single (static)*
type: *integer*

FOB button index.

min	max
1	10

SETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Set Area 1.

SETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Set Area 2.

SETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Set Area 3.

SETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Set Area 4.

SETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Set Area 5.

SETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Set Area 6.

SETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Set Area 7.

SETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Set Area 8.

SETM_AREAS

multiplicity: *single (static)*
type: *integer*

Set Areas.

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

insertV.FOBActUnset

direction: *output*

This is the outgoing call for "insertVFOBActUnset" method.

subindex

multiplicity: *single (static)*
type: *integer*

FOB button index.

min	max
1	10

UNSETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Unset Area 1.

UNSETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Unset Area 2.

UNSETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Unset Area 3.

UNSETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Unset Area 4.

UNSETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Unset Area 5.

UNSETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Unset Area 6.

UNSETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Unset Area 7.

UNSETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Unset Area 8.

UNSETM_AREAS

multiplicity: *single (static)*
type: *integer*

Unset Areas.

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

insertV.FOBActTrigger

direction: *output*

This is the outgoing call for "insertVFOBActTrigger" method.

subindex

multiplicity: *single (static)*
type: *integer*

FOB button index.

min	max
1	10

TRIGGERM_INDEX

multiplicity: *single (static)*
type: *integer*

Trigger index.

min	max
1	255

TRIGGERM_STATE

multiplicity: *single (static)*
type: *integer*

Trigger state.

value	symbol
0	CLEAR
1	SET
2	TOGGLE

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

insertV.FOBActPSet1

direction: *output*

This is the outgoing call for "insertVFOBActPSet1" method.

subindex

multiplicity: *single (static)*
type: *integer*

FOB button index.

min	max
1	10

PSET1M_AREAS.1

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 1.

PSET1M_AREAS.2

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 2.

PSET1M_AREAS.3

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 3.

PSET1M_AREAS.4

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 4.

PSET1M_AREAS.5

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 5.

PSET1M_AREAS.6

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 6.

PSET1M_AREAS.7

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 7.

PSET1M_AREAS.8

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 8.

PSET1M_AREAS

multiplicity: *single (static)*
type: *integer*

PartSet1 Areas.

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

insertV.FOBActPSet2

direction: *output*

This is the outgoing call for "insertVFOBActPSet2" method.

subindex

multiplicity: *single (static)*
type: *integer*

FOB button index.

min	max
1	10

PSET2M_AREAS.1

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 1.

PSET2M_AREAS.2

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 2.

PSET2M_AREAS.3

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 3.

PSET2M_AREAS.4

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 4.

PSET2M_AREAS.5

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 5.

PSET2M_AREAS.6

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 6.

PSET2M_AREAS.7

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 7.

PSET2M_AREAS.8

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 8.

PSET2M_AREAS

multiplicity: *single (static)*
type: *integer*

PartSet2 Areas.

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

insertV.FOBActPanic

direction: *output*

This is the outgoing call for "insertVFOBActPanic" method.

subindex

multiplicity: *single (static)*
type: *integer*

FOB button index.

min	max
1	10

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

insertV.FOBActTakePicture

direction: *output*

This is the outgoing call for "insertVFOBTakePicture" method.

subindex

multiplicity: *single (static)*
type: *integer*

FOB button index.

min	max
1	10

CAMERA_ID

multiplicity: *single (static)*
type: *integer*

Camera ID.

min	max
1	128
257	368

CS_NUMBER

multiplicity: *single (static)*
type: *integer*

Central station number.

min	max
1	16

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

select.Camera

direction: *output*

This is the outgoing call for "selectCamera" method.

index

multiplicity: *single (static)*
type: *integer*

Camera index.

min	max
1	368

return.Camera

direction: *input*

This is the return message for "selectCamera" method.

index

multiplicity: *single (static)*
type: *integer*

Camera index.

min	max
1	368

name

multiplicity: *single (static)*
type: *string*

Camera name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

ZONELIST_ZONE.1

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Zone 1.

min	max
1	128
129	256
257	368

ZONELIST_ZONE.2

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Zone 1.

min	max
1	128
129	256
257	368

ZONELIST_ZONE.3

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Zone 1.

min	max
1	128
129	256
257	368

ZONELIST_ZONE.4

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Zone 1.

min	max
1	128
129	256

min	max
257	368

camFilter1

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

camFilter1.

min	max
1	64

camFilterType1

multiplicity: *single (static)*
type: *integer*

camFilterType1.

value	symbol
0	Burglary
1	Tamper
2	Fault
3	Fire
4	Panic
5	Medical
6	On demand
7	On demand2

camFilterEvent1

multiplicity: *single (static)*
type: *integer*

camFilterEvent1.

value	symbol
9	BC
19	CF
22	CG
25	CL
28	EE
47	HA
56	MA
62	OP
68	PA
80	RX

value	symbol
108	ZA
144	RP
155	OT
156	OK
164	GA
170	KA
176	WA
183	HV
186	UA

camFilter2

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

camFilter1.

min	max
1	64

camFilterType2

multiplicity: *single (static)*
type: *integer*

camFilterType1.

value	symbol
0	Burglary
1	Tamper
2	Fault
3	Fire
4	Panic
5	Medical
6	On demand
7	On demand2

camFilterEvent2

multiplicity: *single (static)*
type: *integer*

camFilterEvent2.

value	symbol
9	BC
19	CF

value	symbol
22	CG
25	CL
28	EE
47	HA
56	MA
62	OP
68	PA
80	RX
108	ZA
144	RP
155	OT
156	OK
164	GA
170	KA
176	WA
183	HV
186	UA

camSIAEvent1

multiplicity: *single (static)*
type: *integer*

camSIAEvent1.

value	symbol
0	NULL
7	BC
15	CF
16	CG
17	CL
18	EE
28	HA
37	MA
43	OP
45	PA
52	RP
56	RX
75	ZA
87	OT
88	OK
91	GA
97	KA
103	WA
111	HV
113	UA

camSIAEvent2

multiplicity: *single (static)*
type: *integer*

camSIAEvent2.

value	symbol
0	NULL
7	BC
15	CF
16	CG
17	CL
18	EE
28	HA
37	MA
43	OP
45	PA
52	RP
56	RX
75	ZA
87	OT
88	OK
91	GA
97	KA
103	WA
111	HV
113	UA

camIsolated

multiplicity: *single (static)*
type: *integer*

camIsolated.

value	symbol
0	false
1	true

camRemoteTrigg

multiplicity: *single (static)*
type: *integer*

camRemoteTrigg.

value	symbol
0	false

value	symbol
1	true

camPictureLimit

multiplicity: *single (static)*
type: *integer*

camPictureLimit.

min	max
0	400

delete.Camera

direction: *output*

This is the outgoing call for "deleteCamera" method.

index

multiplicity: *single (static)*
type: *integer*

Camera index.

min	max
1	368

insert.Camera

direction: *output*

This is the outgoing message for "insertCamera" method.

index

multiplicity: *single (static)*
type: *integer*

Camera index.

min	max
1	368

name

multiplicity: *single (static)*
type: *string*

Camera name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

ZONELIST_ZONE.1

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Zone 1.

min	max
1	128
129	256
257	368

ZONELIST_ZONE.2

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Zone 1.

min	max
1	128
129	256
257	368

ZONELIST_ZONE.3

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Zone 1.

min	max
1	128
129	256
257	368

ZONELIST_ZONE.4

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Zone 1.

min	max
1	128
129	256

min	max
257	368

camFilter1

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

camFilter1.

min	max
1	64

camFilterType1

multiplicity: *single (static)*
type: *integer*

camFilterType1.

value	symbol
0	Burglary
1	Tamper
2	Fault
3	Fire
4	Panic
5	Medical
6	On demand
7	On demand2

camFilterEvent1

multiplicity: *single (static)*
type: *integer*

camFilterEvent1.

value	symbol
9	BC
19	CF
22	CG
25	CL
28	EE
47	HA
56	MA
62	OP
68	PA
80	RX

value	symbol
108	ZA
144	RP
155	OT
156	OK
164	GA
170	KA
176	WA
183	HV
186	UA

camFilter2

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

camFilter1.

min	max
1	64

camFilterType2

multiplicity: *single (static)*
type: *integer*

camFilterType1.

value	symbol
0	Burglary
1	Tamper
2	Fault
3	Fire
4	Panic
5	Medical
6	On demand
7	On demand2

camFilterEvent2

multiplicity: *single (static)*
type: *integer*

camFilterEvent2.

value	symbol
9	BC
19	CF

value	symbol
22	CG
25	CL
28	EE
47	HA
56	MA
62	OP
68	PA
80	RX
108	ZA
144	RP
155	OT
156	OK
164	GA
170	KA
176	WA
183	HV
186	UA

camSIAEvent1

multiplicity: *single (static)*
type: *integer*

camSIAEvent1.

value	symbol
0	NULL
7	BC
15	CF
16	CG
17	CL
18	EE
28	HA
37	MA
43	OP
45	PA
52	RP
56	RX
75	ZA
87	OT
88	OK
91	GA
97	KA
103	WA
111	HV
113	UA

camSIAEvent2

multiplicity: *single (static)*
type: *integer*

camSIAEvent2.

value	symbol
0	NULL
7	BC
15	CF
16	CG
17	CL
18	EE
28	HA
37	MA
43	OP
45	PA
52	RP
56	RX
75	ZA
87	OT
88	OK
91	GA
97	KA
103	WA
111	HV
113	UA

camIsolated

multiplicity: *single (static)*
type: *integer*

camIsolated.

value	symbol
0	false
1	true

camRemoteTrigg

multiplicity: *single (static)*
type: *integer*

camRemoteTrigg.

value	symbol
0	false

value	symbol
1	true

camPictureLimit

multiplicity: *single (static)*
type: *integer*

camPictureLimit.

min	max
0	400

select.DGP

direction: *output*

This is the outgoing call for "selectDGP" method.

index

multiplicity: *single (static)*
type: *integer*

DGP index.

min	max
1	7

return.DGP

direction: *input*

This is the return message for "selectDGP" method.

index

multiplicity: *single (static)*
type: *integer*

DGP index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

DGP name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dgpTampArea

multiplicity: *single (static)*
type: *integer*

dgpTampArea.

min	max
1	8

DGPM_ISOLATE

multiplicity: *single (static)*
type: *integer*

DGPM_ISOLATE.

value	symbol
0	false
1	true

dgpEOL

multiplicity: *single (static)*
type: *integer*

DGPM_EOL

value	symbol
1	10k
2	4k7
3	2k2
4	6k8
5	5k6
6	3k74
7	3k3
8	2k
9	1k5
10	1k
11	8k2
12	4k7+2k2
255	NOEOL

dgpInputMode

multiplicity: *single (static)*
type: *integer*

DGPM_INPUTMODES

value	symbol
0	SINGLENO
1	DUALLOOP
2	SINGLENC

dgpBatteryTestTime

multiplicity: *single (static)*
type: *integer*

Battery test time in minutes.

- The property is available since protocol version 023.
- If the value is equal *255* then the test is performed for unspecified time until the battery is low.

min	max
1	254

value	symbol
255	UNTIL-LOW

insert.DGP

direction: *output*

This is the outgoing message for "insertDGP" method.

index

multiplicity: *single (static)*
type: *integer*

DGP index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

DGP name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dgpTampArea

multiplicity: *single (static)*
type: *integer*

dgpTampArea.

min	max
1	8

DGPM_ISOLATE

multiplicity: *single (static)*
type: *integer*

DGPM_ISOLATE.

value	symbol
0	false
1	true

dgpEOL

multiplicity: *single (static)*
type: *integer*

DGPM_EOL

value	symbol
1	10k
2	4k7
3	2k2
4	6k8
5	5k6
6	3k74
7	3k3
8	2k
9	1k5
10	1k
11	8k2
12	4k7+2k2
255	NOEOL

dgpInputMode

multiplicity: *single (static)*
type: *integer*

DGPM_INPUTMODES

value	symbol
0	SINGLENO
1	DUALLOOP
2	SINGLENC

dgpBatteryTestTime

multiplicity: *single (static)*
type: *integer*

Battery test time in minutes.

- The property is available since protocol version 023.
- If the value is equal *255* then the test is performed for unspecified time until the battery is low.

min	max
1	254

value	symbol
255	UNTIL-LOW

select.Output

direction: *output*

This is the outgoing call for "selectOUTPUT" method.

index

multiplicity: *single (static)*
type: *integer*

OUTPUT index.

min	max
1	200

return.Output

direction: *input*

This is the return message for "selectOUTPUT" method.

index

multiplicity: *single (static)*
type: *integer*

OUTPUT index.

min	max
1	200

name

multiplicity: *single (static)*
type: *string*

Output name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

outInvers

multiplicity: *single (static)*
type: *integer*

outInvers.

value	symbol
0	false
1	true

outFilter

multiplicity: *single (static)*
type: *integer*

outFilter.

min	max
0	64

outInterruptFilter

multiplicity: *single (static)*
type: *integer*

outInterruptFilter.

min	max
0	64

outMode

multiplicity: *single (static)*
type: *integer*

outMode.

value	symbol
0	Follow mode
1	Single Timed mode
2	Double Timed mode
3	Latched mode
4	Toggled mode

outDelayTimeParam

multiplicity: *single (static)*
type: *integer*

outDelayTime.

min	max
0	43200

outActiveTimeParam

multiplicity: *single (static)*
type: *integer*

outActiveTime.

min	max
1	43200

outRetriggerable

multiplicity: *single (static)*
type: *integer*

outRetriggerable.

value	symbol
0	false
1	true

outLogLimit

multiplicity: *single (static)*
type: *integer*

outLogLimit.

value	symbol
0	false
1	true

insert.Output

direction: *output*

This is the outgoing message for "insertOUTPUT" method.

index

multiplicity: *single (static)*
type: *integer*

OUTPUT index.

min	max
1	200

name

multiplicity: *single (static)*
type: *string*

Output name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

outInvers

multiplicity: *single (static)*
type: *integer*

outInvers.

value	symbol
0	false
1	true

outFilter

multiplicity: *single (static)*
type: *integer*

outFilter.

min	max
0	64

outInterruptFilter

multiplicity: *single (static)*
type: *integer*

outInterruptFilter.

min	max
0	64

outMode

multiplicity: *single (static)*
type: *integer*

outMode.

value	symbol
0	Follow mode
1	Single Timed mode
2	Double Timed mode
3	Latched mode
4	Toggled mode

outDelayTimeParam

multiplicity: *single (static)*
type: *integer*

outDelayTime.

min	max
0	43200

outActiveTimeParam

multiplicity: *single (static)*
type: *integer*

outActiveTime.

min	max
1	43200

outRetriggerable

multiplicity: *single (static)*
type: *integer*

outRetriggerable.

value	symbol
0	false
1	true

outLogLimit

multiplicity: *single (static)*
type: *integer*

outLogLimit.

value	symbol
0	false
1	true

select.Trigger

direction: *output*

This is the outgoing call for "selectTrigger" method.

index

multiplicity: *single (static)*
type: *integer*

Trigger index.

min	max
1	255

return.Trigger

direction: *input*

This is the return message for "selectTrigger" method.

index

multiplicity: *single (static)*
type: *integer*

Trigger index.

min	max
1	255

name

multiplicity: *single (static)*
type: *string*

Trigger name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

insert.Trigger

direction: *output*

This is the outgoing message for "insertTrigger" method.

index

multiplicity: *single (static)*
type: *integer*

Trigger index.

min	max
1	255

name

multiplicity: *single (static)*
type: *string*

Trigger name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

select.UserGroup

direction: *output*

This is the outgoing call for "selectUserGroup" method.

index

multiplicity: *single (static)*
type: *integer*

UserGroup index.

min	max
1	16

return.UserGroup

direction: *input*

This is the return message for "selectUserGroup" method.

index

multiplicity: *single (static)*
type: *integer*

UserGroup index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

User group name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

UserGroupArea1

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 1.

UserGroupArea2

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 2.

UserGroupArea3

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 3.

UserGroupArea4

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 4.

UserGroupArea5

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 5.

UserGroupArea6

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 6.

UserGroupArea7

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 7.

UserGroupArea8

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 8.

ugCondFilter

multiplicity: *single (static)*
type: *integer*

ugCondFilter.

min	max
0	64

ugUserGroupType

multiplicity: *single (static)*
type: *integer*

ugUserGroupType.

value	symbol
0	Normal User
1	Supervisor
2	Installer
3	Guard

UGM_OPT_FULLSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_FULLSET.

value	symbol
0	false
1	true

UGM_OPT_PARTSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_PARTSET.

value	symbol
0	false

value	symbol
1	true

UGM_OPT_UNSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_UNSET.

value	symbol
0	false
1	true

UGM_OPT_INHIBIT

multiplicity: *single (static)*
type: *integer*

UGM_OPT_INHIBIT.

value	symbol
0	false
1	true

UGM_OPT_ISOLATE

multiplicity: *single (static)*
type: *integer*

UGM_OPT_ISOLATE.

value	symbol
0	false
1	true

UGM_OPT_TIMEDATE

multiplicity: *single (static)*
type: *integer*

UGM_OPT_TIMEDATE.

value	symbol
0	false
1	true

UGM_OPT_CUSER

multiplicity: *single (static)*
type: *integer*

UGM_OPT_CUSER.

value	symbol
0	UGP_NCUSER
1	UGP_RCUSER
2	UGP_FCUSER

UGM_OPT_FSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_FSET.

value	symbol
0	false
1	true

UGM_OPT_CHGPIN

multiplicity: *single (static)*
type: *integer*

UGM_OPT_CHGPIN.

value	symbol
0	false
1	true

UGM_OPT_WALK

multiplicity: *single (static)*
type: *integer*

UGM_OPT_WALK.

value	symbol
0	false
1	true

UGM_OPT_ENGRESET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_ENGRESET.

value	symbol
0	false
1	true

UGM_OPT_DURESS

multiplicity: *single (static)*
type: *integer*

UGM_OPT_DURESS.

value	symbol
0	false
1	true

UGM_OPT_TESTREP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_TESTREP.

value	symbol
0	false
1	true

UGM_OPT_COMM

multiplicity: *single (static)*
type: *integer*

UGM_OPT_COMM.

value	symbol
0	false
1	true

UGM_OPT_CLEANER

multiplicity: *single (static)*
type: *integer*

UGM_OPT_CLEANER.

value	symbol
0	false
1	true

UGM_AREA_LIST

multiplicity: *single (static)*
type: *integer*

UGM_AREA_LIST.

value	symbol
0	false
1	true

UGM_CP_MODE

multiplicity: *single (static)*
type: *integer*

User Group Card/PIN mode.

value	symbol
0	Card or PIN
1	PIN only
2	Card only

UGM_OPT_MENUACC

multiplicity: *single (static)*
type: *integer*

UGM_OPT_MENUACC.

value	symbol
0	false
1	true

UGM_OPT_INSTACC

multiplicity: *single (static)*
type: *integer*

UGM_OPT_INSTACC.

value	symbol
0	false
1	true

UGM_OPT_VSTOP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_VSTOP.

value	symbol
0	false
1	true

UGM_OPT_LOGSACC

multiplicity: *single (static)*
type: *integer*

UGM_OPT_LOGSACC.

value	symbol
0	false
1	true

UGM_OPT_SMSREP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_SMSREP

value	symbol
0	false
1	true

UGM_OPT_SMSCTRL

multiplicity: *single (static)*
type: *integer*

UGM_OPT_SMSCTRL

value	symbol
0	false
1	true

UGM_OPT_NOPCLREP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_NOPCLREP

value	symbol
0	false
1	true

UGM_OPT_DOORACCESS

multiplicity: *single (static)*
type: *integer*

UGM_OPT_DOORACCESS

value	symbol
0	false
1	true

UGM_OPT_SCHDLMODE

multiplicity: *single (static)*
type: *integer*

UGM_OPT_SCHDLMODE

value	symbol
0	None
1	View
2	View and Control

UGM_OPT_FOBS

multiplicity: *single (static)*
type: *integer*

UGM_OPT_FOBS

value	symbol
0	false
1	true

UGM_OPT_PARTSET2

multiplicity: *single (static)*
type: *integer*

Flag indicating whether the user has privilege to execute second part set.

value	symbol
0	false
1	true

UserGroupRas1

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 1.

UserGroupRas2

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 2.

UserGroupRas3

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 3.

UserGroupRas4

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 4.

UserGroupRas5

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 5.

UserGroupRas6

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 6.

UserGroupRas7

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 7.

UserGroupRas8

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 8.

UGM_OPT_PICT_PRIVMODE

multiplicity: *single (static)*
type: *integer*

Flag indicating whether the user has privilege to Enable/Disable remote picture tiggering.

UGM_OPT_PICT_DELETION

multiplicity: *single (static)*
type: *integer*

Flag indicating whether the user has privilege to execute picture deletion.

insert.UserGroup

direction: *output*

This is the outgoing message for "insertUserGroup" method.

index

multiplicity: *single (static)*
type: *integer*

UserGroup index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

User group name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

UserGroupArea1

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 1.

UserGroupArea2

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 2.

UserGroupArea3

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 3.

UserGroupArea4

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 4.

UserGroupArea5

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 5.

UserGroupArea6

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 6.

UserGroupArea7

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 7.

UserGroupArea8

multiplicity: *single (static)*
type: *boolean*

UserGroupArea 8.

ugCondFilter

multiplicity: *single (static)*
type: *integer*

ugCondFilter.

min	max
0	64

ugUserGroupType

multiplicity: *single (static)*
type: *integer*

ugUserGroupType.

value	symbol
0	Normal User
1	Supervisor
2	Installer
3	Guard

UGM_OPT_FULLSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_FULLSET.

value	symbol
0	false
1	true

UGM_OPT_PARTSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_PARTSET.

value	symbol
0	false

value	symbol
1	true

UGM_OPT_UNSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_UNSET.

value	symbol
0	false
1	true

UGM_OPT_INHIBIT

multiplicity: *single (static)*
type: *integer*

UGM_OPT_INHIBIT.

value	symbol
0	false
1	true

UGM_OPT_ISOLATE

multiplicity: *single (static)*
type: *integer*

UGM_OPT_ISOLATE.

value	symbol
0	false
1	true

UGM_OPT_TIMEDATE

multiplicity: *single (static)*
type: *integer*

UGM_OPT_TIMEDATE.

value	symbol
0	false
1	true

UGM_OPT_CUSER

multiplicity: *single (static)*
type: *integer*

UGM_OPT_CUSER.

value	symbol
0	UGP_NCUSER
1	UGP_RCUSER
2	UGP_FCUSER

UGM_OPT_FSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_FSET.

value	symbol
0	false
1	true

UGM_OPT_CHGPIN

multiplicity: *single (static)*
type: *integer*

UGM_OPT_CHGPIN.

value	symbol
0	false
1	true

UGM_OPT_WALK

multiplicity: *single (static)*
type: *integer*

UGM_OPT_WALK.

value	symbol
0	false
1	true

UGM_OPT_ENGRESET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_ENGRESET.

value	symbol
0	false
1	true

UGM_OPT_DURESS

multiplicity: *single (static)*
type: *integer*

UGM_OPT_DURESS.

value	symbol
0	false
1	true

UGM_OPT_TESTREP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_TESTREP.

value	symbol
0	false
1	true

UGM_OPT_COMM

multiplicity: *single (static)*
type: *integer*

UGM_OPT_COMM.

value	symbol
0	false
1	true

UGM_OPT_CLEANER

multiplicity: *single (static)*
type: *integer*

UGM_OPT_CLEANER.

value	symbol
0	false
1	true

UGM_AREA_LIST

multiplicity: *single (static)*
type: *integer*

UGM_AREA_LIST.

value	symbol
0	false
1	true

UGM_CP_MODE

multiplicity: *single (static)*
type: *integer*

User Group Card/PIN mode.

value	symbol
0	Card or PIN
1	PIN only
2	Card only

UGM_OPT_MENUACC

multiplicity: *single (static)*
type: *integer*

UGM_OPT_MENUACC.

value	symbol
0	false
1	true

UGM_OPT_INSTACC

multiplicity: *single (static)*
type: *integer*

UGM_OPT_INSTACC.

value	symbol
0	false
1	true

UGM_OPT_VSTOP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_VSTOP.

value	symbol
0	false
1	true

UGM_OPT_LOGSACC

multiplicity: *single (static)*
type: *integer*

UGM_OPT_LOGSACC

value	symbol
0	false
1	true

UGM_OPT_SMSREP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_SMSREP

value	symbol
0	false
1	true

UGM_OPT_SMSCTRL

multiplicity: *single (static)*
type: *integer*

UGM_OPT_SMSCTRL

value	symbol
0	false
1	true

UGM_OPT_NOPCLREP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_NOPCLREP

value	symbol
0	false
1	true

UGM_OPT_DOORACCESS

multiplicity: *single (static)*
type: *integer*

UGM_OPT_DOORACCESS

value	symbol
0	false
1	true

UGM_OPT_SCHDLMODE

multiplicity: *single (static)*
type: *integer*

UGM_OPT_SCHDLMODE

value	symbol
0	None
1	View
2	View and Control

UGM_OPT_FOBS

multiplicity: *single (static)*
type: *integer*

UGM_OPT_FOBS

value	symbol
0	false
1	true

UGM_OPT_PARTSET2

multiplicity: *single (static)*
type: *integer*

Flag indicating whether the user has privilege to execute second part set.

value	symbol
0	false
1	true

UserGroupRas1

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 1.

UserGroupRas2

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 2.

UserGroupRas3

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 3.

UserGroupRas4

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 4.

UserGroupRas5

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 5.

UserGroupRas6

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 6.

UserGroupRas7

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 7.

UserGroupRas8

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 8.

UGM_OPT_PICT_PRIVMODE

multiplicity: *single (static)*
type: *integer*

Flag indicating whether the user has privilege to Enable/Disable remote picture tiggering.

UGM_OPT_PICT_DELETION

multiplicity: *single (static)*
type: *integer*

Flag indicating whether the user has privilege to execute picture deletion.

add.UserGroups

direction: *output*

This is the outgoing message for "addUserGroups" method. Use this method for sync.

index

multiplicity: *single (static)*
type: *integer*

UserGroup index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

User group name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

UserGroupArea1

multiplicity: *single (static)*
type: *boolean*

UserGroupArea1.

UserGroupArea2

multiplicity: *single (static)*
type: *boolean*

UserGroupArea2.

UserGroupArea3

multiplicity: *single (static)*
type: *boolean*

UserGroupArea3.

UserGroupArea4

multiplicity: *single (static)*
type: *boolean*

UserGroupArea4.

UserGroupArea5

multiplicity: *single (static)*
type: *boolean*

UserGroupArea5.

UserGroupArea6

multiplicity: *single (static)*
type: *boolean*

UserGroupArea6.

UserGroupArea7

multiplicity: *single (static)*
type: *boolean*

UserGroupArea7.

UserGroupArea8

multiplicity: *single (static)*
type: *boolean*

UserGroupArea8.

ugCondFilter

multiplicity: *single (static)*
type: *integer*

ugCondFilter.

min	max
0	64

ugUserGroupType

multiplicity: *single (static)*
type: *integer*

ugUserGroupType.

value	symbol
0	Normal User
1	Supervisor
2	Installer
3	Guard

UGM_OPT_FULLSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_FULLSET.

value	symbol
0	false
1	true

UGM_OPT_PARTSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_PARTSET.

value	symbol
0	false

value	symbol
1	true

UGM_OPT_UNSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_UNSET.

value	symbol
0	false
1	true

UGM_OPT_INHIBIT

multiplicity: *single (static)*
type: *integer*

UGM_OPT_INHIBIT.

value	symbol
0	false
1	true

UGM_OPT_ISOLATE

multiplicity: *single (static)*
type: *integer*

UGM_OPT_ISOLATE.

value	symbol
0	false
1	true

UGM_OPT_TIMEDATE

multiplicity: *single (static)*
type: *integer*

UGM_OPT_TIMEDATE.

value	symbol
0	false
1	true

UGM_OPT_CUSER

multiplicity: *single (static)*
type: *integer*

UGM_OPT_CUSER.

value	symbol
0	UGP_NCUSER
1	UGP_RCUSER
2	UGP_FCUSER

UGM_OPT_FSET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_FSET.

value	symbol
0	false
1	true

UGM_OPT_CHGPIN

multiplicity: *single (static)*
type: *integer*

UGM_OPT_CHGPIN.

value	symbol
0	false
1	true

UGM_OPT_WALK

multiplicity: *single (static)*
type: *integer*

UGM_OPT_WALK.

value	symbol
0	false
1	true

UGM_OPT_ENGRESET

multiplicity: *single (static)*
type: *integer*

UGM_OPT_ENGRESET.

value	symbol
0	false
1	true

UGM_OPT_DURESS

multiplicity: *single (static)*
type: *integer*

UGM_OPT_DURESS.

value	symbol
0	false
1	true

UGM_OPT_TESTREP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_TESTREP.

value	symbol
0	false
1	true

UGM_OPT_COMM

multiplicity: *single (static)*
type: *integer*

UGM_OPT_COMM.

value	symbol
0	false
1	true

UGM_OPT_CLEANER

multiplicity: *single (static)*
type: *integer*

UGM_OPT_CLEANER.

value	symbol
0	false
1	true

UGM_AREA_LIST

multiplicity: *single (static)*
type: *integer*

UGM_AREA_LIST.

value	symbol
0	false
1	true

UGM_CP_MODE

multiplicity: *single (static)*
type: *integer*

User Group Card/PIN mode.

value	symbol
0	Card or PIN
1	PIN only
2	Card only

UGM_OPT_MENUACC

multiplicity: *single (static)*
type: *integer*

UGM_OPT_MENUACC.

value	symbol
0	false
1	true

UGM_OPT_INSTACC

multiplicity: *single (static)*
type: *integer*

UGM_OPT_INSTACC.

value	symbol
0	false
1	true

UGM_OPT_VSTOP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_VSTOP.

value	symbol
0	false
1	true

UGM_OPT_LOGSACC

multiplicity: *single (static)*
type: *integer*

UGM_OPT_LOGSACC

value	symbol
0	false
1	true

UGM_OPT_SMSREP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_SMSREP

value	symbol
0	false
1	true

UGM_OPT_SMSCTRL

multiplicity: *single (static)*
type: *integer*

UGM_OPT_SMSCTRL

value	symbol
0	false
1	true

UGM_OPT_NOPCLREP

multiplicity: *single (static)*
type: *integer*

UGM_OPT_NOPCLREP

value	symbol
0	false
1	true

UGM_OPT_DOORACCESS

multiplicity: *single (static)*
type: *integer*

UGM_OPT_DOORACCESS

value	symbol
0	false
1	true

UGM_OPT_SCHDLMODE

multiplicity: *single (static)*
type: *integer*

UGM_OPT_SCHDLMODE

value	symbol
0	None
1	View
2	View and Control

UGM_OPT_FOBS

multiplicity: *single (static)*
type: *integer*

UGM_OPT_FOBS

value	symbol
0	false
1	true

UGM_OPT_PARTSET2

multiplicity: *single (static)*
type: *integer*

Flag indicating whether the user has privilege to execute second part set.

value	symbol
0	false
1	true

UserGroupRas1

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 1.

UserGroupRas2

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 2.

UserGroupRas3

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 3.

UserGroupRas4

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 4.

UserGroupRas5

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 5.

UserGroupRas6

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 6.

UserGroupRas7

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 7.

UserGroupRas8

multiplicity: *single (static)*
type: *boolean*

UserGroupRas 8.

UGM_OPT_PICT_PRIVMODE

multiplicity: *single (static)*
type: *integer*

Flag indicating whether the user has privilege to Enable/Disable remote picture tiggering.

UGM_OPT_PICT_DELETION

multiplicity: *single (static)*
type: *integer*

Flag indicating whether the user has privilege to execute picture deletion.

select.DL

direction: *output*

This is the outgoing call for "selectDialer" method.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

selectV.DL

direction: *output*

This is the outgoing call for "selectVDialer" method.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

selectV.DL_MMS

direction: *output*

This is the outgoing call for "selectVDialer" method.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

select.DL_INFO

direction: *output*

This is the outgoing call to receive hardware information about selected dialer. The expected response is `return.DL_INFO` or fault if the device is offline.

See also

- `return.DL_INFO`

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

return.DL_INFO

direction: *input*

This is the return message for `select.DL_INFO` request containing hardware information about the dialer.

See also

- `select.DL_INFO`

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

type

multiplicity: *single (static)*
type: *integer*

Hardware type.

value	symbol
1	BuildIn
2	Dgp

address

multiplicity: *single (static)*
type: *integer*

Hardware address if available.

version

multiplicity: *single (static)*
type: *string*

Hardware version.

return.DL

direction: *input*

This is the return message for "selectDialer" method.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlDevID

multiplicity: *single (static)*
type: *integer*

Dialer dev ID.

min	max
0	255

dlMSN

multiplicity: *single (static)*
type: *string*

MSN phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	false
1	true

dlDialing

multiplicity: *single (static)*
type: *integer*

Dialing otption Pulse/DTMF

value	symbol
0	Pulse
1	DTMF

dlDialTone

multiplicity: *single (static)*
type: *integer*

Dialing otption Pulse/DTMF

value	symbol
0	None
1	Default
3	UK
4	Other

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

return.DL_PSTN

direction: *input*

This is the return message for "selectDialer" method, for PSTN dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

dlEncrypt

multiplicity: *single (static)*
type: *integer*

Encryption ON/OFF.

value	symbol
0	false
1	true

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

dlDialTone

multiplicity: *single (static)*
type: *integer*

Dialing option Pulse/DTMF

value	symbol
0	None
1	Default
3	UK
4	Other

dlDialing

multiplicity: *single (static)*
type: *integer*

Dialing otption Pulse/DTMF

value	symbol
0	Pulse
1	DTMF

dlLFDelay

multiplicity: *single (static)*
type: *integer*

PSTN Line Fault detection delay.

min	max
0	255

return.DL_ISDN

direction: *input*

This is the return message for "selectDialer" method, for ISDN dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

dlEncrypt

multiplicity: *single (static)*
type: *integer*

Encryption ON/OFF.

value	symbol
0	false
1	true

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

dlMSN

multiplicity: *single (static)*
type: *string*

MSN phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

dlPTP

multiplicity: *single (static)*
type: *integer*

If true - the Point to Point mode is selected. If false - the Point to Multipoint mode is selected.

value	symbol
0	false
1	true

return.DL_GSM

direction: *input*

This is the return message for "selectDialer" method, for GSM dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

dlEncrypt

multiplicity: *single (static)*
type: *integer*

Encryption ON/OFF.

value	symbol
0	false
1	true

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

DLM_GSMNET

multiplicity: *single (static)*
type: *string*

GSM network code - empty or 5-6 digits

DLM_PINCODE

multiplicity: *single (static)*
type: *string*

SIM card PIN code: 4 digits

DLM_SMSCNUM

multiplicity: *single (static)*
type: *string*

SMS center phone number. Empty, or GSM number starting with '+'

DLM_CREDITCODE

multiplicity: *single (static)*
type: *string*

Credit check code

DLM_CREDITPERIOD

multiplicity: *single (static)*
type: *integer*

Credit check period in days. If 0 - automatic credit check disabled.

min	max
0	99

DLM_CREDITTIME

multiplicity: *single (static)*
type: *integer*

Credit check time (minutes since 00:00)

min	max
0	1439

DLM_MAXSMSMSGS

multiplicity: *single (static)*
type: *integer*

Max. number of SMS reports during 24hours.

Also: Max. number of SMS messages from unknown sources, forwarded to supervisor during 24hours.

If 0 - no check is performed (unlimited reports/forwards).

min	max
0	200

DLM_SMSHEADER

multiplicity: *single (static)*
type: *string*

SMS report header text.

DLM_SMSFORWARD

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Index of the user to forward unknown SMS messages and automatic credit checks. If 0 - forwarding disabled.

The user selected must be installer, and must belong to the group with SMS Control enabled.

min	max
1	50

DLM_GSMNETMODE

multiplicity: *single (static)*
type: *integer*

If true - only the selected network may be connected by GSM module.

value	symbol
0	false
1	true

DLM_SMSPINREQ

multiplicity: *single (static)*
type: *integer*

If true - user PIN is required at the start of SMS command message.

value	symbol
0	false
1	true

DLM_SMSEXTCHARSET

multiplicity: *single (static)*
type: *integer*

If true - extended character set (UTF16) is allowed in SMS reports and command responses.

value	symbol
0	false
1	true

DLM_CREDITMODE

multiplicity: *single (static)*
type: *integer*

Method of credit check.

value	symbol
0	NONE
1	CODE
2	SMS

DLM_CREDITNUM

multiplicity: *single (static)*
type: *string*

Phone number for SMS-mode credit check. Empty, or GSM number starting with '+'

return.DL_IP

direction: *input*

This is the return message for "selectDialer" method, for IP dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

dlEncrypt

multiplicity: *single (static)*
type: *integer*

Encryption ON/OFF.

value	symbol
0	false
1	true

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

DLM_USEDHCP

multiplicity: *single (static)*
type: *integer*

Use DHCP YES/NO

value	symbol
0	false
1	true

DLM_AUTODNS

multiplicity: *single (static)*
type: *integer*

Get DNS from DHCP YES/NO

value	symbol
0	false
1	true

DLM_AUTONTP

multiplicity: *single (static)*
type: *integer*

Get NTP from DHCP YES/NO

value	symbol
0	false
1	true

DLM_USEFIREWALL

multiplicity: *single (static)*
type: *integer*

Firewall ON/OFF

value	symbol
0	false
1	true

DLM_REPLYPING

multiplicity: *single (static)*
type: *integer*

Replay on PING ON/OFF

value	symbol
0	false
1	true

DLM_IPADDR

multiplicity: *single (static)*
type: *string*

IP address

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_NETMASK

multiplicity: *single (static)*
type: *string*

Netmask

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_ROUTER

multiplicity: *single (static)*
type: *string*

Gateway

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_DNSSERVER

multiplicity: *single (static)*
type: *string*

DNS server

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_NTPSERVER

multiplicity: *single (static)*
type: *string*

DNS server

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_ETHSPEED

multiplicity: *single (static)*
type: *integer*

Ethernet link speed

value	symbol
0	AUTO
1	10MB
2	100MB

DLM_MAXETHPICMSGS

multiplicity: *single (static)*
type: *integer*

Limit for reported pictures per 24h and arm-disarm cycle

min	max
0	200

return.DL_STEL

direction: *input*

This is the return message for "selectDialer" method, for Safetel dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

return.DL_CHIRON

direction: *input*

This is the return message for "selectDialer" method, for TDA74xx dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

return.DL_75XX

direction: *input*

This is the return message for "selectDialer" method, for ATS75XX dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

return.DL_VEMPTY

direction: *input*

Dummy return message for variant part of the non GSM dialer configuration.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

return.DL_GSM_IP

direction: *input*

This is the return message for "selectDialer" method, for GSM dialer, IP part.

DLM_APNNAME

multiplicity: *single (static)*
type: *string*

Access Point Name

A computer protocol that allows panel to access the Internet using the mobile phone network.

DLM_USERNAME

multiplicity: *single (static)*
type: *string*

Specific user name defined by GPRS provider.

DLM_USERPASS

multiplicity: *single (static)*
type: *string*

Specific user password defined by GPRS provider.

DLM_PPPTIMEOUT

multiplicity: *single (static)*
type: *integer*

Timeout value 5 min - 23 h : 59 min, value 23 h : 59 min means permanent connection.

min	**max**
5	1439

DLM_AUTODNS

multiplicity: *single (static)*
type: *integer*

Get DNS from DHCP YES/NO.

value	symbol
0	false
1	true

DLM_USEFIREWALL

multiplicity: *single (static)*
type: *integer*

Firewall ON/OFF

value	symbol
0	false
1	true

DLM_REPLYPING

multiplicity: *single (static)*
type: *integer*

Replay on PING ON/OFF

value	symbol
0	false
1	true

DLM_DNSSERVER

multiplicity: *single (static)*
type: *string*

DNS server

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_PPPLF

multiplicity: *single (static)*
type: *integer*

PPP Line Fault

value	symbol
0	false
1	true

DLM_MAXGPRSPICMSGS

multiplicity: *single (static)*
type: *integer*

Picture limit for GPRS link

min	max
0	200

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

return.DL_GSM_MMS

direction: *input*

This is the return message for "selectDialer" method, for GSM dialer, MMS part.

DLM_MMS_APN_NAME

multiplicity: *single (static)*
type: *string*

Access Point Name

A computer protocol that allows panel to send MMSs using the mobile phone network.

DLM_MMS_USER_NAME

multiplicity: *single (static)*
type: *string*

Specific user name defined by GSM provider.

DLM_MMS_USER_PASSWORD

multiplicity: *single (static)*
type: *string*

Specific user password defined by GSM provider.

DLM_MMS_SERVER_NAME

multiplicity: *single (static)*
type: *string*

MMS Central address.

DLM_MMS_PROXYADDR

multiplicity: *single (static)*
type: *string*

Proxy address for MMS Central..

DLM_MMS_PROXYPORT

multiplicity: *single (static)*
type: *integer*

Proxy port for MMS Central.

DLM_MMS_MAXMSGS

multiplicity: *single (static)*
type: *integer*

Picture limit for MMS link

min	max
0	200

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

insert.DL

direction: *output*

This is the outgoing message for "insertDialer" method.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlDevID

multiplicity: *single (static)*
type: *integer*

Dialer dev ID.

min	max
0	255

dlMSN

multiplicity: *single (static)*
type: *string*

MSN phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	false
1	true

dlDialing

multiplicity: *single (static)*
type: *integer*

Dialing otption Pulse/DTMF

value	symbol
0	Pulse
1	DTMF

dlDialTone

multiplicity: *single (static)*
type: *integer*

Dialing tone selection

value	symbol
0	None
1	Default
3	UK
4	Other

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

select.SiaEvent

direction: *output*

This is the outgoing call for "selectSiaEvent" method.

index

multiplicity: *single (static)*
type: *integer*

Sia event index.

value	symbol
1	AN
2	AR
3	AS
4	AT
5	BA
6	BB
7	BC
8	BJ
9	BR
10	BT
11	BU
12	BV
13	BW
14	BZ
15	CF
16	CG
17	CL
18	EE
19	ER
20	ET
21	FA
22	FB
23	FJ
24	FR
25	FT
26	FU
27	FW
28	HA
29	HR
30	JP
31	JR

value	symbol
32	JT
33	LB
34	LR
35	LS
36	LT
37	MA
38	MB
39	MJ
40	MR
41	MS
42	MU
43	OP
44	OR
45	PA
46	PB
47	PJ
48	PR
49	PT
50	PU
51	RB
52	RP
53	RR
54	RS
55	RU
56	RX
57	TA
58	TB
59	TR
60	TT
61	TU
62	UB
63	UU
64	WF
65	WP
66	XH
67	XQ
68	XR
69	XT
70	YC
71	YK
72	YR
73	YS
74	YT
75	ZA
76	ZB

value	symbol
77	ZJ
78	ZR
79	ZS
80	ZU
81	YA
82	YH
83	NC
84	NR
85	CP
86	OA
87	OT
88	OK
89	IA
90	IR
91	GA
92	GR
93	GB
94	GU
95	GS
96	GJ
97	KA
98	KR
99	KB
100	KU
101	KS
102	KJ
103	WA
104	WR
105	WB
106	WU
107	WS
108	WJ
109	ES
110	EJ
111	HV
112	HW
113	UA
114	UR
115	TS
116	TE
117	LU
118	YP
119	YQ

insert.SiaEvent

direction: *output*

This is the outgoing message for "insertSiaEvent" method.

index

multiplicity: *single (static)*
type: *integer*

Sia event index.

value	symbol
1	AN
2	AR
3	AS
4	AT
5	BA
6	BB
7	BC
8	BJ
9	BR
10	BT
11	BU
12	BV
13	BW
14	BZ
15	CF
16	CG
17	CL
18	EE
19	ER
20	ET
21	FA
22	FB
23	FJ
24	FR
25	FT
26	FU
27	FW
28	HA
29	HR
30	JP
31	JR

value	symbol
32	JT
33	LB
34	LR
35	LS
36	LT
37	MA
38	MB
39	MJ
40	MR
41	MS
42	MU
43	OP
44	OR
45	PA
46	PB
47	PJ
48	PR
49	PT
50	PU
51	RB
52	RP
53	RR
54	RS
55	RU
56	RX
57	TA
58	TB
59	TR
60	TT
61	TU
62	UB
63	UU
64	WF
65	WP
66	XH
67	XQ
68	XR
69	XT
70	YC
71	YK
72	YR
73	YS
74	YT
75	ZA
76	ZB

value	symbol
77	ZJ
78	ZR
79	ZS
80	ZU
81	YA
82	YH
83	NC
84	NR
85	CP
86	OA
87	OT
88	OK
89	IA
90	IR
91	GA
92	GR
93	GB
94	GU
95	GS
96	GJ
97	KA
98	KR
99	KB
100	KU
101	KS
102	KJ
103	WA
104	WR
105	WB
106	WU
107	WS
108	WJ
109	ES
110	EJ
111	HV
112	HW
113	UA
114	UR
115	TS
116	TE
117	LU
118	YP
119	YQ

rep_to_cs_1

multiplicity: *single (static)*
type: *integer*

Report to CS 1.

value	symbol
0	false
1	true

rep_to_cs_2

multiplicity: *single (static)*
type: *integer*

Report to CS 2.

value	symbol
0	false
1	true

rep_to_cs_3

multiplicity: *single (static)*
type: *integer*

Report to CS 3.

value	symbol
0	false
1	true

rep_to_cs_4

multiplicity: *single (static)*
type: *integer*

Report to CS 4.

value	symbol
0	false
1	true

rep_to_cs_5

multiplicity: *single (static)*
type: *integer*

Report to CS 5.

value	symbol
0	false
1	true

rep_to_cs_6

multiplicity: *single (static)*
type: *integer*

Report to CS 6.

value	symbol
0	false
1	true

rep_to_cs_7

multiplicity: *single (static)*
type: *integer*

Report to CS 7.

value	symbol
0	false
1	true

rep_to_cs_8

multiplicity: *single (static)*
type: *integer*

Report to CS 8.

value	symbol
0	false
1	true

rep_to_cs_9

multiplicity: *single (static)*
type: *integer*

Report to CS 9.

value	symbol
0	false
1	true

rep_to_cs_10

multiplicity: *single (static)*
type: *integer*

Report to CS 10.

value	symbol
0	false
1	true

rep_to_cs_11

multiplicity: *single (static)*
type: *integer*

Report to CS 11.

value	symbol
0	false
1	true

rep_to_cs_12

multiplicity: *single (static)*
type: *integer*

Report to CS 12.

value	symbol
0	false
1	true

rep_to_cs_13

multiplicity: *single (static)*
type: *integer*

Report to CS 13.

value	symbol
0	false
1	true

rep_to_cs_14

multiplicity: *single (static)*
type: *integer*

Report to CS 14.

value	symbol
0	false
1	true

rep_to_cs_15

multiplicity: *single (static)*
type: *integer*

Report to CS 15.

value	symbol
0	false
1	true

rep_to_cs_16

multiplicity: *single (static)*
type: *integer*

Report to CS 16.

value	symbol
0	false
1	true

opt_DELAY

multiplicity: *single (static)*
type: *integer*

Delay Enable/Disable

value	symbol
0	false
1	true

voice_msg_no

multiplicity: *single (static)*
type: *integer*

Voice message number.

min	max
0	14

return.SiaEvent

direction: *input*

This is the return message for "selectSiaEvent" method.

index

multiplicity: *single (static)*
type: *integer*

Sia event index.

value	symbol
1	AN
2	AR
3	AS
4	AT
5	BA
6	BB
7	BC
8	BJ
9	BR
10	BT
11	BU
12	BV
13	BW
14	BZ
15	CF
16	CG
17	CL
18	EE
19	ER
20	ET
21	FA
22	FB
23	FJ
24	FR
25	FT
26	FU
27	FW
28	HA
29	HR
30	JP
31	JR

value	symbol
32	JT
33	LB
34	LR
35	LS
36	LT
37	MA
38	MB
39	MJ
40	MR
41	MS
42	MU
43	OP
44	OR
45	PA
46	PB
47	PJ
48	PR
49	PT
50	PU
51	RB
52	RP
53	RR
54	RS
55	RU
56	RX
57	TA
58	TB
59	TR
60	TT
61	TU
62	UB
63	UU
64	WF
65	WP
66	XH
67	XQ
68	XR
69	XT
70	YC
71	YK
72	YR
73	YS
74	YT
75	ZA
76	ZB

value	symbol
77	ZJ
78	ZR
79	ZS
80	ZU
81	YA
82	YH
83	NC
84	NR
85	CP
86	OA
87	OT
88	OK
89	IA
90	IR
91	GA
92	GR
93	GB
94	GU
95	GS
96	GJ
97	KA
98	KR
99	KB
100	KU
101	KS
102	KJ
103	WA
104	WR
105	WB
106	WU
107	WS
108	WJ
109	ES
110	EJ
111	HV
112	HW
113	UA
114	UR
115	TS
116	TE
117	LU
118	YP
119	YQ

rep_to_cs_1

multiplicity: *single (static)*
type: *integer*

Report to CS 1.

value	symbol
0	false
1	true

rep_to_cs_2

multiplicity: *single (static)*
type: *integer*

Report to CS 2.

value	symbol
0	false
1	true

rep_to_cs_3

multiplicity: *single (static)*
type: *integer*

Report to CS 3.

value	symbol
0	false
1	true

rep_to_cs_4

multiplicity: *single (static)*
type: *integer*

Report to CS 4.

value	symbol
0	false
1	true

rep_to_cs_5

multiplicity: *single (static)*
type: *integer*

Report to CS 5.

value	symbol
0	false
1	true

rep_to_cs_6

multiplicity: *single (static)*
type: *integer*

Report to CS 6.

value	symbol
0	false
1	true

rep_to_cs_7

multiplicity: *single (static)*
type: *integer*

Report to CS 7.

value	symbol
0	false
1	true

rep_to_cs_8

multiplicity: *single (static)*
type: *integer*

Report to CS 8.

value	symbol
0	false
1	true

rep_to_cs_9

multiplicity: *single (static)*
type: *integer*

Report to CS 9.

value	symbol
0	false
1	true

rep_to_cs_10

multiplicity: *single (static)*
type: *integer*

Report to CS 10.

value	symbol
0	false
1	true

rep_to_cs_11

multiplicity: *single (static)*
type: *integer*

Report to CS 11.

value	symbol
0	false
1	true

rep_to_cs_12

multiplicity: *single (static)*
type: *integer*

Report to CS 12.

value	symbol
0	false
1	true

rep_to_cs_13

multiplicity: *single (static)*
type: *integer*

Report to CS 13.

value	symbol
0	false
1	true

rep_to_cs_14

multiplicity: *single (static)*
type: *integer*

Report to CS 14.

value	symbol
0	false
1	true

rep_to_cs_15

multiplicity: *single (static)*
type: *integer*

Report to CS 15.

value	symbol
0	false
1	true

rep_to_cs_16

multiplicity: *single (static)*
type: *integer*

Report to CS 16.

value	symbol
0	false
1	true

opt_DELAY

multiplicity: *single (static)*
type: *integer*

Delay Enable/Disable

value	symbol
0	false
1	true

voice_msg_no

multiplicity: *single (static)*
type: *integer*

Voice message number.

min	max
0	14

select.PCC

direction: *output*

This is the outgoing call for "selectPCC" method.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

selectV.PCC

direction: *output*

This is the outgoing call for "selectPCC" method, variant part.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

return.PCC

direction: *input*

This is the return message for "selectPCC" method.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

PC connection name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

pccPhoneNumber

multiplicity: *single (static)*
type: *string*

PCC phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

pccDataLink

multiplicity: *single (static)*
type: *integer*

Data link type.

min	max
1	4
6	6

pccModemProtocol

multiplicity: *single (static)*
type: *integer*

Data link type.

value	symbol
0	MODP_V21
1	MODP_V22
2	MODP_V22BIS
3	MODP_BELL103
4	MODP_CSD
5	MODP_GPRS

pccRetryLimit

multiplicity: *single (static)*
type: *integer*

Retry limit.

min	max
0	250

return.PCC_2

direction: *input*

This is the return message for "selectPCC" method.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

PC connection name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

pccDataLink

multiplicity: *single (static)*
type: *integer*

Data link type.

min	max
1	4
6	6

pccModemProtocol

multiplicity: *single (static)*
type: *integer*

Data link type.

value	symbol
0	MODP_V21
1	MODP_V22
2	MODP_V22BIS
3	MODP_BELL103
4	MODP_CSD
5	MODP_GPRS

pccRetryLimit

multiplicity: *single (static)*
type: *integer*

Retry limit.

min	max
0	250

pccPhoneNumber

multiplicity: *single (static)*
type: *string*

PCC phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

PCCM_IP_ADDRESS

multiplicity: *single (static)*
type: *string*

IP Address of central station.

PCCM_IP_PORT

multiplicity: *single (static)*
type: *integer*

PCC IP port

return.PCC_CMN

direction: *input*

This is the return message for "selectPCC" method.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

PC connection name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

pccDataLink

multiplicity: *single (static)*
type: *integer*

Data link type.

min	max
1	4
6	6

pccModemProtocol

multiplicity: *single (static)*
type: *integer*

Modem prot. type.

value	symbol
0	MODP_V21
1	MODP_V22
2	MODP_V22BIS
3	MODP_BELL103
4	MODP_CSD
5	MODP_GPRS

pccRetryLimit

multiplicity: *single (static)*
type: *integer*

Retry limit.

min	max
0	250

return.PCC_PHONE

direction: *input*

This is the return message for method "selectPCC",variant part with PHONE num

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

phoneNum

multiplicity: *single (static)*
type: *string*

Phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters:: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

return.PCC_IP

direction: *input*

This is the return message for method "selectPCC",variant part with IP addr

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

PCCM_IP_ADDRESS

multiplicity: *single (static)*
type: *string*

IP Address of PC.

PCCM_IP_PORT

multiplicity: *single (static)*
type: *integer*

PC IP port

insert.PCC

direction: *output*

This is the outgoing message for "insertPCC" method.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

PC connection name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

pccPhoneNumber

multiplicity: *single (static)*
type: *string*

PCC phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

pccDataLink

multiplicity: *single (static)*
type: *integer*

Data link type.

min	max
1	4
6	6

pccModemProtocol

multiplicity: *single (static)*
type: *integer*

Data link type.

value	symbol
0	MODP_V21
1	MODP_V22
2	MODP_V22BIS
3	MODP_BELL103
4	MODP_CSD
5	MODP_GPRS

pccRetryLimit

multiplicity: *single (static)*
type: *integer*

Retry limit.

min	max
0	250

insert.PCC_2

direction: *output*

This is the outgoing message for "insertPCC" method.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

PC connection name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

pccDataLink

multiplicity: *single (static)*
type: *integer*

Data link type.

min	max
1	4
6	6

pccModemProtocol

multiplicity: *single (static)*
type: *integer*

Data link type.

value	symbol
0	MODP_V21
1	MODP_V22
2	MODP_V22BIS
3	MODP_BELL103
4	MODP_CSD
5	MODP_GPRS

pccRetryLimit

multiplicity: *single (static)*
type: *integer*

Retry limit.

min	max
0	250

pccPhoneNumber

multiplicity: *single (static)*
type: *string*

PCC phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

PCCM_IP_ADDRESS

multiplicity: *single (static)*
type: *string*

IP Address of central station.

PCCM_IP_PORT

multiplicity: *single (static)*
type: *integer*

PCC IP port

insert.PCC_CMN

direction: *output*

This is the outgoing message for "insertPCC" method.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

name

multiplicity: *single (static)*
type: *string*

PC connection name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

pccDataLink

multiplicity: *single (static)*
type: *integer*

Data link type.

min	max
1	4
6	6

pccModemProtocol

multiplicity: *single (static)*
type: *integer*

Modem prot. type.

value	symbol
0	MODP_V21
1	MODP_V22
2	MODP_V22BIS
3	MODP_BELL103
4	MODP_CSD
5	MODP_GPRS

pccRetryLimit

multiplicity: *single (static)*
type: *integer*

Retry limit.

min	max
0	250

insertV.PCC_PHONE

direction: *output*

This is the outgoing message for "insertPCC" method, variant part with PHONE num.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

phoneNum

multiplicity: *single (static)*
type: *string*

PCC phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

insertV.PCC_IP

direction: *output*

This is the outgoing message for "insertPCC" method, variant part with IP addr.

index

multiplicity: *single (static)*
type: *integer*

PC Connections index.

min	max
1	16

PCCM_IP_ADDRESS

multiplicity: *single (static)*
type: *string*

IP Address of central station.

PCCM_IP_PORT

multiplicity: *single (static)*
type: *integer*

PCC IP port

return.CommandStatus

direction: *input*

This is the return message for every "insert" method. It returns TRUE if successful

commandStatus

multiplicity: *single (static)*
type: *integer*

Status of last "insert" command

value	symbol
0	false
1	true

select.CEvFilter

direction: *output*

This is the outgoing call for "selectCEvFilter" method.

index

multiplicity: *single (static)*
type: *integer*

CEvFilter index.

min	max
1	64

return.CEvFilter

direction: *input*

This is the return message for "selectCEvFilter" method.

name

multiplicity: *single (static)*
type: *string*

CEvFilter name.

invers

multiplicity: *single (static)*
type: *integer*

Inverted output.

event1

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Event number.

event1.zone

multiplicity: *single (static)*
type: *integer*

Zone Event number.

Notes:

- ZNEV_LEARNED is available since protocol version 018.
- ZNEV_PRELEARNED is available since protocol version 020.
- ZNEV_HELDOPEN and ZNEV_INVWT_INACTIVE are available since protocol version 023.

value	symbol
1	ZNEV_ACTIVE
2	ZNEV_TAMPER
3	ZNEV_AM
4	ZNEV_BATTFAIL

value	symbol
5	ZNEV_FAULT
6	ZNEV_DIRTY
7	ZNEV_SVSHORT
8	ZNEV_SVLONG
9	ZNEV_INHIBIT
10	ZNEV_ISOLATE
11	ZNEV_SOAK
12	ZNEV_SET
13	ZNEV_ALARM
14	ZNEV_LEARNED
15	ZNEV_PRELEARNED
16	ZNEV_HELDOPEN
17	ZNEV_INVWT_INACTIVE

event1.area

multiplicity: *single (static)*
type: *integer*

Area Event number.

value	symbol
1	AREV_FULLSET
2	AREV_PARTSET
3	AREV_UNSET
4	AREV_ALARM
5	AREV_FSALARM
6	AREV_PSALARM
7	AREV_USALARM
8	AREV_FTCALARM
9	AREV_FIREDOOR
10	AREV_FSFIREDOOR
11	AREV_PSFIREDOOR
12	AREV_USFIREDOOR
13	AREV_FTCFIREDOOR
14	AREV_FIRE
15	AREV_FSFIRE
16	AREV_PSFIRE
17	AREV_USFIRE
18	AREV_FTCFIRE
19	AREV_PANIC
20	AREV_FSPANIC
21	AREV_PSPANIC
22	AREV_USPANIC
23	AREV_FTCPANIC
24	AREV_MEDICAL

value	symbol
25	AREV_FSMEDICAL
26	AREV_PSMEDICAL
27	AREV_USMEDICAL
28	AREV_FTCMEDICAL
29	AREV_TECHNICAL
30	AREV_FSTECHNICAL
31	AREV_PSTECHNICAL
32	AREV_USTECHNICAL
33	AREV_FTCTECHNICAL
34	AREV_TAMPER
35	AREV_FSTAMPER
36	AREV_PSTAMPER
37	AREV_USTAMPER
38	AREV_FTCTAMPER
39	AREV_DOORBELL
40	AREV_PSDOORBEL
41	AREV_USDOORBEL
42	AREV_ZNACTIVE
43	AREV_ZNINHIBIT
44	AREV_ZNISOLATE
45	AREV_ZNFAULT
46	AREV_ZNAM
47	AREV_ZNTAMPER
48	AREV_RASTAMPER
49	AREV_RASFAULT
50	AREV_DGPTAMPER
51	AREV_DGPFAULT
52	AREV_DURESS
53	AREV_FSDURESS
54	AREV_PSDURESS
55	AREV_USDURESS
56	AREV_FTCDURESS
57	AREV_CODETAMPER
58	AREV_ENTRY
59	AREV_EXIT
60	AREV_EXITFAULT
61	AREV_RTS
62	AREV_SETOK
63	AREV_SETFAULT
64	AREV_UNSETOK
65	AREV_ALARMACK
66	AREV_FIRERESET
67	AREV_WALK
68	AREV_WALKZNACTV
69	AREV_AALARM

value	symbol
70	AREV_BALARM
71	AREV_ISIREN
72	AREV_ESIREN
73	AREV_STROBE
74	AREV_BUZZER
75	AREV_AMRESET
76	AREV_PARTSET2
77	AREV_WARNING
78	AREV_AUTOARM
79	AREV_HAALARM
80	AREV_HBALARM

event1.ras

multiplicity: *single (static)*
type: *integer*

RAS Event number.

value	symbol
1	RASEV_OFFLINE
2	RASEV_RTE
3	RASEV_CODETAMPER
4	RASEV_TAMPER
5	RASEV_DURESS
6	RASEV_CARD
7	RASEV_PIN
8	RASEV_DOORACC
9	RASEV_LOCKED
10	RASEV_ISOLATE
11	RASEV_DOORBELL
12	RASEV_CARDV
13	RASEV_EXIT_START
14	RASEV_ENTRY_STOPPED

event1.dgp

multiplicity: *single (static)*
type: *integer*

DGP Event number.

value	symbol
1	DGPEV_OFFLINE
2	DGPEV_MAINSFAIL
3	DGPEV_BATTFAIL
4	DGPEV_TAMPER

value	symbol
5	DGPEV_FUSEFAULT
6	DGPEV_SIRENFAULT
7	DGPEV_RCVFAULT
8	DGPEV_ISOLATE
9	DGPEV_BATTLOW
10	DGPEV_BTESTACTV
11	DGPEV_BTESTFAIL
12	DGPEV_PU_FAIL

event1.panel

multiplicity: *single (static)*
type: *integer*

Panel Event number.

value	symbol
1	DGP0EV_MAINSFAIL
2	DGP0EV_BATTFAIL
3	DGP0EV_TAMPER
4	DGP0EV_FUSEFAULT
5	DGP0EV_SIRENFAULT
6	DGP0EV_LF
7	DGP0EV_LFPSTN
8	DGP0EV_LFISDN
9	DGP0EV_LFGSM
10	DGP0EV_FTC
11	DGP0EV_MIFAULT
12	DGP0EV_MIFISDN
13	DGP0EV_MIFGSM
14	DGP0EV_MIFVOICE
15	DGP0EV_NTPF
16	DGP0EV_LFETH
17	DGP0EV_LFIP
18	DGP0EV_LFGPRS
19	DGP0EV_LFIPGPRS
20	DGP0EV_LFTDA
21	DGP0EV_LFTDAGPRS
22	DGP0EV_LFTDAETH
23	DGP0EV_MIFTDA

event1.user

multiplicity: *single (static)*
type: *integer*

User Event number.

value	symbol
1	USREV_CARDPIN
2	USREV_SMSCTRLACTIVE
3	USREV_SMSCTRLLOCK
4	USREV_SMSREPACTIVE
5	USREV_SMSREPAFTERSET

event1.output

multiplicity: *single (static)*
type: *integer*

Output Event number.

value	symbol
1	OUTEV_ACTIVE
2	OUTEV_ON

event1.filter

multiplicity: *single (static)*
type: *integer*

Zone Event number.

value	symbol
1	CFLEV_ACTIVE

event1.system

multiplicity: *single (static)*
type: *integer*

System Event number.

value	symbol
1	SYSEV_ALLSET
2	SYSEV_AUTOANS
3	SYSEV_RCONNACTV
4	SYSEV_RCONNFAIL
5	SYSEV_LPRGACTV
6	SYSEV_RPRGACTV
7	SYSEV_TIMECHG
8	SYSEV_SSAVER
9	SYSEV_ISIREN
10	SYSEV_ESIREN
11	SYSEV_STROBE
12	SYSEV_FAULT
13	SYSEV_TAMPER

value	symbol
14	SYSEV_SERVICEIN

event1.trigger

multiplicity: *single (static)*
type: *integer*

Trigger Event number.

value	symbol
1	TRGEV_KEYFOBSW1
2	TRGEV_KEYFOBSW2
3	TRGEV_KEYFOBSW12
4	TRGEV_REMOTEOUT
5	TRGEV_FKEY
6	TRGEV_SCHEDULE
7	TRGEV_FOB

event1.cs

multiplicity: *single (static)*
type: *integer*

Central Station Event number.

value	symbol
1	CSEV_FTC
2	CSEV_HBF
3	CSEV_BUSY

event1.ug

multiplicity: *single (static)*
type: *integer*

User Group Event number.

value	symbol
1	UGEV_CARDPIN

event1.sexc

multiplicity: *single (static)*
type: *integer*

Schedule Exception Event number.

value	symbol
1	EXCPEV_ACTIVE

event1.scal

multiplicity: *single (static)*
type: *integer*

Schedule Calendar Event number.

value	symbol
1	SCALEV_HOUR
2	SCALEV_DAY
3	SCALEV_MON
4	SCALEV_TUE
5	SCALEV_WED
6	SCALEV_THU
7	SCALEV_FRI
8	SCALEV_SAT
9	SCALEV_SUN

event1.fob

multiplicity: *single (static)*
type: *integer*

Fob Event number.

value	symbol
1	FOBEV_LEARNED
2	FOBEV_BUTTON1
3	FOBEV_BUTTON2
4	FOBEV_BUTTON3
5	FOBEV_BUTTON4
6	FOBEV_BUTTON12
7	FOBEV_BUTTON13
8	FOBEV_BUTTON14
9	FOBEV_BUTTON23
10	FOBEV_BUTTON24
11	FOBEV_BUTTON34

event1.camera

multiplicity: *single (static)*
type: *integer*

Fob Event number.

value	symbol
1	CAMEV_PICTURE_CAPTURED
2	CAMEV_EV_LIMIT_EXCEEDED

class1

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

object class.

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

obj_no1

multiplicity: *single (static)*
type: *integer*

Object number.

obj_no1.zone

multiplicity: *single (static)*
type: *integer*

Zone number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no1.area

multiplicity: *single (static)*
type: *integer*

Area number.

min	max
1	8

value	symbol
0	ANY

obj_no1.ras

multiplicity: *single (static)*
type: *integer*

RAS number.

min	max
1	8

value	symbol
0	ANY

obj_no1.dgp

multiplicity: *single (static)*
type: *integer*

DGP number.

min	max
1	7

value	symbol
0	ANY

obj_no1.user

multiplicity: *single (static)*
type: *integer*

User number.

min	max
1	50

value	symbol
0	ANY

obj_no1.output

multiplicity: *single (static)*
type: *integer*

Output number.

min	max
1	200

value	symbol
0	ANY

obj_no1.filter

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Filter number.

min	max
1	64

obj_no1.trigger

multiplicity: *single (static)*
type: *integer*

Trigger number.

min	max
1	255

value	symbol
0	ANY

obj_no1.ug

multiplicity: *single (static)*
type: *integer*

UG number.

min	max
1	16

value	symbol
0	ANY

obj_no1.sexc

multiplicity: *single (static)*
type: *integer*

Schedule exception number.

min	max
1	64

value	symbol
0	ANY

obj_no1.fob

multiplicity: *single (static)*
type: *integer*

Fob number.

min	max
1	112

value	symbol
0	ANY

obj_no1.camera

multiplicity: *single (static)*
type: *integer*

Camera number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no1_invert

multiplicity: *single (static)*
type: *integer*

Invert

value	symbol
0	false
1	true

event2

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Event number.

event2.zone

multiplicity: *single (static)*
type: *integer*

Zone Event number.

Notes:

- `ZNEV_LEARNED` is available since protocol version `018`.
- `ZNEV_PRELEARNED` is available since protocol version `020`.
- `ZNEV_HELDOPEN` and `ZNEV_INVWT_INACTIVE` are available since protocol version `023`.

value	symbol
1	ZNEV_ACTIVE
2	ZNEV_TAMPER
3	ZNEV_AM
4	ZNEV_BATTFAIL
5	ZNEV_FAULT
6	ZNEV_DIRTY
7	ZNEV_SVSHORT
8	ZNEV_SVLONG
9	ZNEV_INHIBIT
10	ZNEV_ISOLATE
11	ZNEV_SOAK
12	ZNEV_SET
13	ZNEV_ALARM
14	ZNEV_LEARNED
15	ZNEV_PRELEARNED
16	ZNEV_HELDOPEN
17	ZNEV_INVWT_INACTIVE

event2.area

multiplicity: *single (static)*
type: *integer*

Area Event number.

value	symbol
1	AREV_FULLSET
2	AREV_PARTSET
3	AREV_UNSET
4	AREV_ALARM
5	AREV_FSALARM
6	AREV_PSALARM
7	AREV_USALARM
8	AREV_FTCALARM
9	AREV_FIREDOOR
10	AREV_FSFIREDOOR
11	AREV_PSFIREDOOR
12	AREV_USFIREDOOR
13	AREV_FTCFIREDOOR
14	AREV_FIRE
15	AREV_FSFIRE
16	AREV_PSFIRE
17	AREV_USFIRE
18	AREV_FTCFIRE
19	AREV_PANIC
20	AREV_FSPANIC
21	AREV_PSPANIC
22	AREV_USPANIC
23	AREV_FTCPANIC
24	AREV_MEDICAL
25	AREV_FSMEDICAL
26	AREV_PSMEDICAL
27	AREV_USMEDICAL
28	AREV_FTCMEDICAL
29	AREV_TECHNICAL
30	AREV_FSTECHNICAL
31	AREV_PSTECHNICAL
32	AREV_USTECHNICAL
33	AREV_FTCTECHNICAL
34	AREV_TAMPER
35	AREV_FSTAMPER
36	AREV_PSTAMPER
37	AREV_USTAMPER
38	AREV_FTCTAMPER
39	AREV_DOORBELL

value	symbol
40	AREV_PSDOORBEL
41	AREV_USDOORBEL
42	AREV_ZNACTIVE
43	AREV_ZNINHIBIT
44	AREV_ZNISOLATE
45	AREV_ZNFAULT
46	AREV_ZNAM
47	AREV_ZNTAMPER
48	AREV_RASTAMPER
49	AREV_RASFAULT
50	AREV_DGPTAMPER
51	AREV_DGPFAULT
52	AREV_DURESS
53	AREV_FSDURESS
54	AREV_PSDURESS
55	AREV_USDURESS
56	AREV_FTCDURESS
57	AREV_CODETAMPER
58	AREV_ENTRY
59	AREV_EXIT
60	AREV_EXITFAULT
61	AREV_RTS
62	AREV_SETOK
63	AREV_SETFAULT
64	AREV_UNSETOK
65	AREV_ALARMACK
66	AREV_FIRERESET
67	AREV_WALK
68	AREV_WALKZNACTV
69	AREV_AALARM
70	AREV_BALARM
71	AREV_ISIREN
72	AREV_ESIREN
73	AREV_STROBE
74	AREV_BUZZER
75	AREV_AMRESET
76	AREV_PARTSET2
77	AREV_WARNING
78	AREV_AUTOARM
79	AREV_HAALARM
80	AREV_HBALARM

event2.ras

multiplicity: *single (static)*
type: *integer*

RAS Event number.

value	symbol
1	RASEV_OFFLINE
2	RASEV_RTE
3	RASEV_CODETAMPER
4	RASEV_TAMPER
5	RASEV_DURESS
6	RASEV_CARD
7	RASEV_PIN
8	RASEV_DOORACC
9	RASEV_LOCKED
10	RASEV_ISOLATE
11	RASEV_DOORBELL
12	RASEV_CARDV
13	RASEV_EXIT_START
14	RASEV_ENTRY_STOPPED

event2.dgp

multiplicity: *single (static)*
type: *integer*

DGP Event number.

value	symbol
1	DGPEV_OFFLINE
2	DGPEV_MAINSFAIL
3	DGPEV_BATTFAIL
4	DGPEV_TAMPER
5	DGPEV_FUSEFAULT
6	DGPEV_SIRENFAULT
7	DGPEV_RCVFAULT
8	DGPEV_ISOLATE
9	DGPEV_BATTLOW
10	DGPEV_BTESTACTV
11	DGPEV_BTESTFAIL
12	DGPEV_PU_FAIL

event2.panel

multiplicity: *single (static)*
type: *integer*

Panel Event number.

value	symbol
1	DGP0EV_MAINSFAIL
2	DGP0EV_BATTFAIL
3	DGP0EV_TAMPER
4	DGP0EV_FUSEFAULT
5	DGP0EV_SIRENFAULT
6	DGP0EV_LF
7	DGP0EV_LFPSTN
8	DGP0EV_LFISDN
9	DGP0EV_LFGSM
10	DGP0EV_FTC
11	DGP0EV_MIFAULT
12	DGP0EV_MIFISDN
13	DGP0EV_MIFGSM
14	DGP0EV_MIFVOICE
15	DGP0EV_NTPF
16	DGP0EV_LFETH
17	DGP0EV_LFIP
18	DGP0EV_LFGPRS
19	DGP0EV_LFIPGPRS
20	DGP0EV_LFTDA
21	DGP0EV_LFTDAGPRS
22	DGP0EV_LFTDAETH
23	DGP0EV_MIFTDA

event2.user

multiplicity: *single (static)*
type: *integer*

User Event number.

value	symbol
1	USREV_CARDPIN
2	USREV_SMSCTRLACTIVE
3	USREV_SMSCTRLLOCK
4	USREV_SMSREPACTIVE
5	USREV_SMSREPAFTERSET

event2.output

multiplicity: *single (static)*
type: *integer*

Output Event number.

value	symbol
1	OUTEV_ACTIVE
2	OUTEV_ON

event2.filter

multiplicity: *single (static)*
type: *integer*

Zone Event number.

value	symbol
1	CFLEV_ACTIVE

event2.system

multiplicity: *single (static)*
type: *integer*

System Event number.

value	symbol
1	SYSEV_ALLSET
2	SYSEV_AUTOANS
3	SYSEV_RCONNACTV
4	SYSEV_RCONNFAIL
5	SYSEV_LPRGACTV
6	SYSEV_RPRGACTV
7	SYSEV_TIMECHG
8	SYSEV_SSAVER
9	SYSEV_ISIREN
10	SYSEV_ESIREN
11	SYSEV_STROBE
12	SYSEV_FAULT
13	SYSEV_TAMPER
14	SYSEV_SERVICEIN

event2.trigger

multiplicity: *single (static)*
type: *integer*

Trigger Event number.

value	symbol
1	TRGEV_KEYFOBSW1
2	TRGEV_KEYFOBSW2
3	TRGEV_KEYFOBSW12
4	TRGEV_REMOTEOUT
5	TRGEV_FKEY
6	TRGEV_SCHEDULE
7	TRGEV_FOB

event2.sexc

multiplicity: *single (static)*
type: *integer*

Schedule Exception Event number.

value	symbol
1	EXCPEV_ACTIVE

event2.scal

multiplicity: *single (static)*
type: *integer*

Schedule Calendar Event number.

value	symbol
1	SCALEV_HOUR
2	SCALEV_DAY
3	SCALEV_MON
4	SCALEV_TUE
5	SCALEV_WED
6	SCALEV_THU
7	SCALEV_FRI
8	SCALEV_SAT
9	SCALEV_SUN

event2.fob

multiplicity: *single (static)*
type: *integer*

Fob Event number.

value	symbol
1	FOBEV_LEARNED
2	FOBEV_BUTTON1
3	FOBEV_BUTTON2
4	FOBEV_BUTTON3
5	FOBEV_BUTTON4
6	FOBEV_BUTTON12
7	FOBEV_BUTTON13
8	FOBEV_BUTTON14
9	FOBEV_BUTTON23
10	FOBEV_BUTTON24
11	FOBEV_BUTTON34

event2.camera

multiplicity: *single (static)*
type: *integer*

Camera Event number.

value	symbol
1	CAMEV_PICTURE_CAPTURED
2	CAMEV_EV_LIMIT_EXCEEDED

event2.cs

multiplicity: *single (static)*
type: *integer*

Central Station Event number.

value	symbol
1	CSEV_FTC
2	CSEV_HBF
3	CSEV_BUSY

event2.ug

multiplicity: *single (static)*
type: *integer*

User Group Event number.

value	symbol
1	UGEV_CARDPIN

class2

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

object class.

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

obj_no2

multiplicity: *single (static)*
type: *integer*

Object number.

obj_no2.zone

multiplicity: *single (static)*
type: *integer*

Zone number.

min	max
1	128

min	max
257	368

value	symbol
0	ANY

obj_no2.area

multiplicity: *single (static)*
type: *integer*

Area number.

min	max
1	8

value	symbol
0	ANY

obj_no2.ras

multiplicity: *single (static)*
type: *integer*

RAS number.

min	max
1	8

value	symbol
0	ANY

obj_no2.dgp

multiplicity: *single (static)*
type: *integer*

DGP number.

min	max
1	7

value	symbol
0	ANY

obj_no2.user

multiplicity: *single (static)*
type: *integer*

User number.

min	max
1	50

value	symbol
0	ANY

obj_no2.output

multiplicity: *single (static)*
type: *integer*

Output number.

min	max
1	200

value	symbol
0	ANY

obj_no2.filter

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Filter number.

min	max
1	64

obj_no2.trigger

multiplicity: *single (static)*
type: *integer*

Trigger number.

min	max
1	255

value	symbol
0	ANY

obj_no2.ug

multiplicity: *single (static)*
type: *integer*

UG number.

min	max
1	16

value	symbol
0	ANY

obj_no2.sexc

multiplicity: *single (static)*
type: *integer*

Schedule exception number.

min	max
1	64

value	symbol
0	ANY

obj_no2.fob

multiplicity: *single (static)*
type: *integer*

Fob number.

min	max
1	112

value	symbol
0	ANY

obj_no2.camera

multiplicity: *single (static)*
type: *integer*

Camera number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no2_invert

multiplicity: *single (static)*
type: *integer*

Invert

value	symbol
0	false
1	true

event3

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Event number.

event3.zone

multiplicity: *single (static)*
type: *integer*

Zone Event number.

Notes:

- `ZNEV_LEARNED` is available since protocol version `018`.
- `ZNEV_PRELEARNED` is available since protocol version `020`.
- `ZNEV_HELDOPEN` and `ZNEV_INVWT_INACTIVE` are available since protocol version `023`.

value	symbol
1	ZNEV_ACTIVE
2	ZNEV_TAMPER
3	ZNEV_AM
4	ZNEV_BATTFAIL
5	ZNEV_FAULT
6	ZNEV_DIRTY
7	ZNEV_SVSHORT
8	ZNEV_SVLONG
9	ZNEV_INHIBIT
10	ZNEV_ISOLATE
11	ZNEV_SOAK
12	ZNEV_SET
13	ZNEV_ALARM
14	ZNEV_LEARNED
15	ZNEV_PRELEARNED
16	ZNEV_HELDOPEN
17	ZNEV_INVWT_INACTIVE

event3.area

multiplicity: *single (static)*
type: *integer*

Area Event number.

value	symbol
1	AREV_FULLSET
2	AREV_PARTSET
3	AREV_UNSET
4	AREV_ALARM
5	AREV_FSALARM
6	AREV_PSALARM
7	AREV_USALARM
8	AREV_FTCALARM
9	AREV_FIREDOOR
10	AREV_FSFIREDOOR
11	AREV_PSFIREDOOR
12	AREV_USFIREDOOR
13	AREV_FTCFIREDOOR
14	AREV_FIRE
15	AREV_FSFIRE
16	AREV_PSFIRE
17	AREV_USFIRE
18	AREV_FTCFIRE
19	AREV_PANIC
20	AREV_FSPANIC
21	AREV_PSPANIC
22	AREV_USPANIC
23	AREV_FTCPANIC
24	AREV_MEDICAL
25	AREV_FSMEDICAL
26	AREV_PSMEDICAL
27	AREV_USMEDICAL
28	AREV_FTCMEDICAL
29	AREV_TECHNICAL
30	AREV_FSTECHNICAL
31	AREV_PSTECHNICAL
32	AREV_USTECHNICAL
33	AREV_FTCTECHNICAL
34	AREV_TAMPER
35	AREV_FSTAMPER
36	AREV_PSTAMPER
37	AREV_USTAMPER
38	AREV_FTCTAMPER
39	AREV_DOORBELL

value	symbol
40	AREV_PSDOORBEL
41	AREV_USDOORBEL
42	AREV_ZNACTIVE
43	AREV_ZNINHIBIT
44	AREV_ZNISOLATE
45	AREV_ZNFAULT
46	AREV_ZNAM
47	AREV_ZNTAMPER
48	AREV_RASTAMPER
49	AREV_RASFAULT
50	AREV_DGPTAMPER
51	AREV_DGPFAULT
52	AREV_DURESS
53	AREV_FSDURESS
54	AREV_PSDURESS
55	AREV_USDURESS
56	AREV_FTCDURESS
57	AREV_CODETAMPER
58	AREV_ENTRY
59	AREV_EXIT
60	AREV_EXITFAULT
61	AREV_RTS
62	AREV_SETOK
63	AREV_SETFAULT
64	AREV_UNSETOK
65	AREV_ALARMACK
66	AREV_FIRERESET
67	AREV_WALK
68	AREV_WALKZNACTV
69	AREV_AALARM
70	AREV_BALARM
71	AREV_ISIREN
72	AREV_ESIREN
73	AREV_STROBE
74	AREV_BUZZER
75	AREV_AMRESET
76	AREV_PARTSET2
77	AREV_WARNING
78	AREV_AUTOARM
79	AREV_HAALARM
80	AREV_HBALARM

event3.ras

multiplicity: *single (static)*
type: *integer*

RAS Event number.

value	symbol
1	RASEV_OFFLINE
2	RASEV_RTE
3	RASEV_CODETAMPER
4	RASEV_TAMPER
5	RASEV_DURESS
6	RASEV_CARD
7	RASEV_PIN
8	RASEV_DOORACC
9	RASEV_LOCKED
10	RASEV_ISOLATE
11	RASEV_DOORBELL
12	RASEV_CARDV
13	RASEV_EXIT_START
14	RASEV_ENTRY_STOPPED

event3.dgp

multiplicity: *single (static)*
type: *integer*

DGP Event number.

value	symbol
1	DGPEV_OFFLINE
2	DGPEV_MAINSFAIL
3	DGPEV_BATTFAIL
4	DGPEV_TAMPER
5	DGPEV_FUSEFAULT
6	DGPEV_SIRENFAULT
7	DGPEV_RCVFAULT
8	DGPEV_ISOLATE
9	DGPEV_BATTLOW
10	DGPEV_BTESTACTV
11	DGPEV_BTESTFAIL
12	DGPEV_PU_FAIL

event3.panel

multiplicity: *single (static)*
type: *integer*

Panel Event number.

value	symbol
1	DGP0EV_MAINSFAIL
2	DGP0EV_BATTFAIL
3	DGP0EV_TAMPER
4	DGP0EV_FUSEFAULT
5	DGP0EV_SIRENFAULT
6	DGP0EV_LF
7	DGP0EV_LFPSTN
8	DGP0EV_LFISDN
9	DGP0EV_LFGSM
10	DGP0EV_FTC
11	DGP0EV_MIFAULT
12	DGP0EV_MIFISDN
13	DGP0EV_MIFGSM
14	DGP0EV_MIFVOICE
15	DGP0EV_NTPF
16	DGP0EV_LFETH
17	DGP0EV_LFIP
18	DGP0EV_LFGPRS
19	DGP0EV_LFIPGPRS
20	DGP0EV_LFTDA
21	DGP0EV_LFTDAGPRS
22	DGP0EV_LFTDAETH
23	DGP0EV_MIFTDA

event3.user

multiplicity: *single (static)*
type: *integer*

User Event number.

value	symbol
1	USREV_CARDPIN
2	USREV_SMSCTRLACTIVE
3	USREV_SMSCTRLLOCK
4	USREV_SMSREPACTIVE
5	USREV_SMSREPAFTERSET

event3.output

multiplicity: *single (static)*
type: *integer*

Output Event number.

value	symbol
1	OUTEV_ACTIVE
2	OUTEV_ON

event3.filter

multiplicity: *single (static)*
type: *integer*

Zone Event number.

value	symbol
1	CFLEV_ACTIVE

event3.system

multiplicity: *single (static)*
type: *integer*

System Event number.

value	symbol
1	SYSEV_ALLSET
2	SYSEV_AUTOANS
3	SYSEV_RCONNACTV
4	SYSEV_RCONNFAIL
5	SYSEV_LPRGACTV
6	SYSEV_RPRGACTV
7	SYSEV_TIMECHG
8	SYSEV_SSAVER
9	SYSEV_ISIREN
10	SYSEV_ESIREN
11	SYSEV_STROBE
12	SYSEV_FAULT
13	SYSEV_TAMPER
14	SYSEV_SERVICEIN

event3.trigger

multiplicity: *single (static)*
type: *integer*

Trigger Event number.

value	symbol
1	TRGEV_KEYFOBSW1
2	TRGEV_KEYFOBSW2
3	TRGEV_KEYFOBSW12
4	TRGEV_REMOTEOUT
5	TRGEV_FKEY
6	TRGEV_SCHEDULE
7	TRGEV_FOB

event3.cs

multiplicity: *single (static)*
type: *integer*

Central Station Event number.

value	symbol
1	CSEV_FTC
2	CSEV_HBF
3	CSEV_BUSY

event3.ug

multiplicity: *single (static)*
type: *integer*

User Group Event number.

value	symbol
1	UGEV_CARDPIN

event3.sexc

multiplicity: *single (static)*
type: *integer*

Schedule Exception Event number.

value	symbol
1	EXCPEV_ACTIVE

event3.scal

multiplicity: *single (static)*
type: *integer*

Schedule Calendar Event number.

value	symbol
1	SCALEV_HOUR
2	SCALEV_DAY
3	SCALEV_MON
4	SCALEV_TUE
5	SCALEV_WED
6	SCALEV_THU
7	SCALEV_FRI
8	SCALEV_SAT
9	SCALEV_SUN

event3.fob

multiplicity: *single (static)*
type: *integer*

Fob Event number.

value	symbol
1	FOBEV_LEARNED
2	FOBEV_BUTTON1
3	FOBEV_BUTTON2
4	FOBEV_BUTTON3
5	FOBEV_BUTTON4
6	FOBEV_BUTTON12
7	FOBEV_BUTTON13
8	FOBEV_BUTTON14
9	FOBEV_BUTTON23
10	FOBEV_BUTTON24
11	FOBEV_BUTTON34

event3.camera

multiplicity: *single (static)*
type: *integer*

Camera Event number.

value	symbol
1	CAMEV_PICTURE_CAPTURED
2	CAMEV_EV_LIMIT_EXCEEDED

class3

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

object class.

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

obj_no3

multiplicity: *single (static)*
type: *integer*

Object number.

obj_no3.zone

multiplicity: *single (static)*
type: *integer*

Zone number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no3.area

multiplicity: *single (static)*
type: *integer*

Area number.

min	max
1	8

value	symbol
0	ANY

obj_no3.ras

multiplicity: *single (static)*
type: *integer*

RAS number.

min	max
1	8

value	symbol
0	ANY

obj_no3.dgp

multiplicity: *single (static)*
type: *integer*

DGP number.

min	max
1	7

value	symbol
0	ANY

obj_no3.user

multiplicity: *single (static)*
type: *integer*

User number.

min	max
1	50

value	symbol
0	ANY

obj_no3.output

multiplicity: *single (static)*
type: *integer*

Output number.

min	max
1	200

value	symbol
0	ANY

obj_no3.filter

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Filter number.

min	max
1	64

obj_no3.trigger

multiplicity: *single (static)*
type: *integer*

Trigger number.

min	max
1	255

value	symbol
0	ANY

obj_no3.ug

multiplicity: *single (static)*
type: *integer*

UG number.

min	max
1	16

value	symbol
0	ANY

obj_no3.sexc

multiplicity: *single (static)*
type: *integer*

Schedule exception number.

min	max
1	64

value	symbol
0	ANY

obj_no3.fob

multiplicity: *single (static)*
type: *integer*

Fob number.

min	max
1	112

value	symbol
0	ANY

obj_no3.camera

multiplicity: *single (static)*
type: *integer*

Camera number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no3_invert

multiplicity: *single (static)*
type: *integer*

Invert

value	symbol
0	false
1	true

event4

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Event number.

event4.zone

multiplicity: *single (static)*
type: *integer*

Zone Event number.

Notes:

- ZNEV_LEARNED is available since protocol version 018.
- ZNEV_PRELEARNED is available since protocol version 020.
- ZNEV_HELDOPEN and ZNEV_INVWT_INACTIVE are available since protocol version 023.

value	symbol
1	ZNEV_ACTIVE
2	ZNEV_TAMPER
3	ZNEV_AM
4	ZNEV_BATTFAIL
5	ZNEV_FAULT
6	ZNEV_DIRTY
7	ZNEV_SVSHORT
8	ZNEV_SVLONG
9	ZNEV_INHIBIT
10	ZNEV_ISOLATE
11	ZNEV_SOAK
12	ZNEV_SET
13	ZNEV_ALARM
14	ZNEV_LEARNED
15	ZNEV_PRELEARNED
16	ZNEV_HELDOPEN
17	ZNEV_INVWT_INACTIVE

event4.area

multiplicity: *single (static)*
type: *integer*

Area Event number.

value	symbol
1	AREV_FULLSET
2	AREV_PARTSET
3	AREV_UNSET
4	AREV_ALARM
5	AREV_FSALARM
6	AREV_PSALARM
7	AREV_USALARM
8	AREV_FTCALARM
9	AREV_FIREDOOR
10	AREV_FSFIREDOOR
11	AREV_PSFIREDOOR
12	AREV_USFIREDOOR
13	AREV_FTCFIREDOOR
14	AREV_FIRE
15	AREV_FSFIRE
16	AREV_PSFIRE
17	AREV_USFIRE
18	AREV_FTCFIRE
19	AREV_PANIC
20	AREV_FSPANIC
21	AREV_PSPANIC
22	AREV_USPANIC
23	AREV_FTCPANIC
24	AREV_MEDICAL
25	AREV_FSMEDICAL
26	AREV_PSMEDICAL
27	AREV_USMEDICAL
28	AREV_FTCMEDICAL
29	AREV_TECHNICAL
30	AREV_FSTECHNICAL
31	AREV_PSTECHNICAL
32	AREV_USTECHNICAL
33	AREV_FTCTECHNICAL
34	AREV_TAMPER
35	AREV_FSTAMPER
36	AREV_PSTAMPER
37	AREV_USTAMPER
38	AREV_FTCTAMPER
39	AREV_DOORBELL

value	symbol
40	AREV_PSDOORBEL
41	AREV_USDOORBEL
42	AREV_ZNACTIVE
43	AREV_ZNINHIBIT
44	AREV_ZNISOLATE
45	AREV_ZNFAULT
46	AREV_ZNAM
47	AREV_ZNTAMPER
48	AREV_RASTAMPER
49	AREV_RASFAULT
50	AREV_DGPTAMPER
51	AREV_DGPFAULT
52	AREV_DURESS
53	AREV_FSDURESS
54	AREV_PSDURESS
55	AREV_USDURESS
56	AREV_FTCDURESS
57	AREV_CODETAMPER
58	AREV_ENTRY
59	AREV_EXIT
60	AREV_EXITFAULT
61	AREV_RTS
62	AREV_SETOK
63	AREV_SETFAULT
64	AREV_UNSETOK
65	AREV_ALARMACK
66	AREV_FIRERESET
67	AREV_WALK
68	AREV_WALKZNACTV
69	AREV_AALARM
70	AREV_BALARM
71	AREV_ISIREN
72	AREV_ESIREN
73	AREV_STROBE
74	AREV_BUZZER
75	AREV_AMRESET
76	AREV_PARTSET2
77	AREV_WARNING
78	AREV_AUTOARM
79	AREV_HAALARM
80	AREV_HBALARM

event4.ras

multiplicity: *single (static)*
type: *integer*

RAS Event number.

value	symbol
1	RASEV_OFFLINE
2	RASEV_RTE
3	RASEV_CODETAMPER
4	RASEV_TAMPER
5	RASEV_DURESS
6	RASEV_CARD
7	RASEV_PIN
8	RASEV_DOORACC
9	RASEV_LOCKED
10	RASEV_ISOLATE
11	RASEV_DOORBELL
12	RASEV_CARDV
13	RASEV_EXIT_START
14	RASEV_ENTRY_STOPPED

event4.dgp

multiplicity: *single (static)*
type: *integer*

DGP Event number.

value	symbol
1	DGPEV_OFFLINE
2	DGPEV_MAINSFAIL
3	DGPEV_BATTFAIL
4	DGPEV_TAMPER
5	DGPEV_FUSEFAULT
6	DGPEV_SIRENFAULT
7	DGPEV_RCVFAULT
8	DGPEV_ISOLATE
9	DGPEV_BATTLOW
10	DGPEV_BTESTACTV
11	DGPEV_BTESTFAIL
12	DGPEV_PU_FAIL

event4.panel

multiplicity: *single (static)*
type: *integer*

Panel Event number.

value	symbol
1	DGP0EV_MAINSFAIL
2	DGP0EV_BATTFAIL
3	DGP0EV_TAMPER
4	DGP0EV_FUSEFAULT
5	DGP0EV_SIRENFAULT
6	DGP0EV_LF
7	DGP0EV_LFPSTN
8	DGP0EV_LFISDN
9	DGP0EV_LFGSM
10	DGP0EV_FTC
11	DGP0EV_MIFAULT
12	DGP0EV_MIFISDN
13	DGP0EV_MIFGSM
14	DGP0EV_MIFVOICE
15	DGP0EV_NTPF
16	DGP0EV_LFETH
17	DGP0EV_LFIP
18	DGP0EV_LFGPRS
19	DGP0EV_LFIPGPRS
20	DGP0EV_LFTDA
21	DGP0EV_LFTDAGPRS
22	DGP0EV_LFTDAETH
23	DGP0EV_MIFTDA

event4.user

multiplicity: *single (static)*
type: *integer*

User Event number.

value	symbol
1	USREV_CARDPIN
2	USREV_SMSCTRLACTIVE
3	USREV_SMSCTRLLOCK
4	USREV_SMSREPACTIVE
5	USREV_SMSREPAFTERSET

event4.output

multiplicity: *single (static)*
type: *integer*

Output Event number.

value	symbol
1	OUTEV_ACTIVE
2	OUTEV_ON

event4.filter

multiplicity: *single (static)*
type: *integer*

Zone Event number.

value	symbol
1	CFLEV_ACTIVE

event4.system

multiplicity: *single (static)*
type: *integer*

System Event number.

value	symbol
1	SYSEV_ALLSET
2	SYSEV_AUTOANS
3	SYSEV_RCONNACTV
4	SYSEV_RCONNFAIL
5	SYSEV_LPRGACTV
6	SYSEV_RPRGACTV
7	SYSEV_TIMECHG
8	SYSEV_SSAVER
9	SYSEV_ISIREN
10	SYSEV_ESIREN
11	SYSEV_STROBE
12	SYSEV_FAULT
13	SYSEV_TAMPER
14	SYSEV_SERVICEIN

event4.trigger

multiplicity: *single (static)*
type: *integer*

Trigger Event number.

value	symbol
1	TRGEV_KEYFOBSW1
2	TRGEV_KEYFOBSW2
3	TRGEV_KEYFOBSW12
4	TRGEV_REMOTEOUT
5	TRGEV_FKEY
6	TRGEV_SCHEDULE
7	TRGEV_FOB

event4.cs

multiplicity: *single (static)*
type: *integer*

Central Station Event number.

value	symbol
1	CSEV_FTC
2	CSEV_HBF
3	CSEV_BUSY

event4.ug

multiplicity: *single (static)*
type: *integer*

User Group Event number.

value	symbol
1	UGEV_CARDPIN

event4.sexc

multiplicity: *single (static)*
type: *integer*

Schedule Exception Event number.

value	symbol
1	EXCPEV_ACTIVE

event4.scal

multiplicity: *single (static)*
type: *integer*

Schedule Calendar Event number.

value	symbol
1	SCALEV_HOUR
2	SCALEV_DAY
3	SCALEV_MON
4	SCALEV_TUE
5	SCALEV_WED
6	SCALEV_THU
7	SCALEV_FRI
8	SCALEV_SAT
9	SCALEV_SUN

event4.fob

multiplicity: *single (static)*
type: *integer*

Fob Event number.

value	symbol
1	FOBEV_LEARNED
2	FOBEV_BUTTON1
3	FOBEV_BUTTON2
4	FOBEV_BUTTON3
5	FOBEV_BUTTON4
6	FOBEV_BUTTON12
7	FOBEV_BUTTON13
8	FOBEV_BUTTON14
9	FOBEV_BUTTON23
10	FOBEV_BUTTON24
11	FOBEV_BUTTON34

event4.camera

multiplicity: *single (static)*
type: *integer*

Camera Event number.

value	symbol
1	CAMEV_PICTURE_CAPTURED
2	CAMEV_EV_LIMIT_EXCEEDED

class4

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

object class.

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

obj_no4

multiplicity: *single (static)*
type: *integer*

Object number.

obj_no4.zone

multiplicity: *single (static)*
type: *integer*

Zone number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no4.area

multiplicity: *single (static)*
type: *integer*

Area number.

min	max
1	8

value	symbol
0	ANY

obj_no4.ras

multiplicity: *single (static)*
type: *integer*

RAS number.

min	max
1	8

value	symbol
0	ANY

obj_no4.dgp

multiplicity: *single (static)*
type: *integer*

DGP number.

min	max
1	7

value	symbol
0	ANY

obj_no4.user

multiplicity: *single (static)*
type: *integer*

User number.

min	max
1	50

value	symbol
0	ANY

obj_no4.output

multiplicity: *single (static)*
type: *integer*

Output number.

min	max
1	200

value	symbol
0	ANY

obj_no4.filter

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Filter number.

min	max
1	64

obj_no4.trigger

multiplicity: *single (static)*
type: *integer*

Trigger number.

min	max
1	255

value	symbol
0	ANY

obj_no4.ug

multiplicity: *single (static)*
type: *integer*

UG number.

min	max
1	16

value	symbol
0	ANY

obj_no4.sexc

multiplicity: *single (static)*
type: *integer*

Schedule exception number.

min	max
1	64

value	symbol
0	ANY

obj_no4.fob

multiplicity: *single (static)*
type: *integer*

Fob number.

min	max
1	112

value	symbol
0	ANY

obj_no4.camera

multiplicity: *single (static)*
type: *integer*

Camera number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no4_invert

multiplicity: *single (static)*
type: *integer*

Invert

value	symbol
0	false
1	true

op1

multiplicity: *single (static)*
type: *integer*

operator 1.

value	symbol
0	OR
1	AND
2	XOR

op2

multiplicity: *single (static)*
type: *integer*

operator 2.

value	symbol
0	OR
1	AND
2	XOR

op3

multiplicity: *single (static)*
type: *integer*

operator 3.

value	symbol
0	OR
1	AND
2	XOR

index

multiplicity: *single (static)*
type: *integer*

CEvFilter index.

min	max
1	64

insert.CEvFilter

direction: *output*

This is the outgoing message for "insertCEvFilter" method.

name

multiplicity: *single (static)*
type: *string*

CEvFilter name.

invers

multiplicity: *single (static)*
type: *integer*

Inverted output.

event1

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Event number.

event1.zone

multiplicity: *single (static)*
type: *integer*

Zone Event number.

Notes:

- ZNEV_LEARNED is available since protocol version 018.
- ZNEV_PRELEARNED is available since protocol version 020.
- ZNEV_HELDOPEN and ZNEV_INVWT_INACTIVE are available since protocol version 023.

value	symbol
1	ZNEV_ACTIVE
2	ZNEV_TAMPER
3	ZNEV_AM
4	ZNEV_BATTFAIL

value	symbol
5	ZNEV_FAULT
6	ZNEV_DIRTY
7	ZNEV_SVSHORT
8	ZNEV_SVLONG
9	ZNEV_INHIBIT
10	ZNEV_ISOLATE
11	ZNEV_SOAK
12	ZNEV_SET
13	ZNEV_ALARM
14	ZNEV_LEARNED
15	ZNEV_PRELEARNED
16	ZNEV_HELDOPEN
17	ZNEV_INVWT_INACTIVE

event1.area

multiplicity: *single (static)*
type: *integer*

Area Event number.

value	symbol
1	AREV_FULLSET
2	AREV_PARTSET
3	AREV_UNSET
4	AREV_ALARM
5	AREV_FSALARM
6	AREV_PSALARM
7	AREV_USALARM
8	AREV_FTCALARM
9	AREV_FIREDOOR
10	AREV_FSFIREDOOR
11	AREV_PSFIREDOOR
12	AREV_USFIREDOOR
13	AREV_FTCFIREDOOR
14	AREV_FIRE
15	AREV_FSFIRE
16	AREV_PSFIRE
17	AREV_USFIRE
18	AREV_FTCFIRE
19	AREV_PANIC
20	AREV_FSPANIC
21	AREV_PSPANIC
22	AREV_USPANIC
23	AREV_FTCPANIC
24	AREV_MEDICAL

value	symbol
25	AREV_FSMEDICAL
26	AREV_PSMEDICAL
27	AREV_USMEDICAL
28	AREV_FTCMEDICAL
29	AREV_TECHNICAL
30	AREV_FSTECHNICAL
31	AREV_PSTECHNICAL
32	AREV_USTECHNICAL
33	AREV_FTCTECHNICAL
34	AREV_TAMPER
35	AREV_FSTAMPER
36	AREV_PSTAMPER
37	AREV_USTAMPER
38	AREV_FTCTAMPER
39	AREV_DOORBELL
40	AREV_PSDOORBEL
41	AREV_USDOORBEL
42	AREV_ZNACTIVE
43	AREV_ZNINHIBIT
44	AREV_ZNISOLATE
45	AREV_ZNFAULT
46	AREV_ZNAM
47	AREV_ZNTAMPER
48	AREV_RASTAMPER
49	AREV_RASFAULT
50	AREV_DGPTAMPER
51	AREV_DGPFAULT
52	AREV_DURESS
53	AREV_FSDURESS
54	AREV_PSDURESS
55	AREV_USDURESS
56	AREV_FTCDURESS
57	AREV_CODETAMPER
58	AREV_ENTRY
59	AREV_EXIT
60	AREV_EXITFAULT
61	AREV_RTS
62	AREV_SETOK
63	AREV_SETFAULT
64	AREV_UNSETOK
65	AREV_ALARMACK
66	AREV_FIRERESET
67	AREV_WALK
68	AREV_WALKZNACTV
69	AREV_AALARM

value	symbol
70	AREV_BALARM
71	AREV_ISIREN
72	AREV_ESIREN
73	AREV_STROBE
74	AREV_BUZZER
75	AREV_AMRESET
76	AREV_PARTSET2
77	AREV_WARNING
78	AREV_AUTOARM
79	AREV_HAALARM
80	AREV_HBALARM

event1.ras

multiplicity: *single (static)*
type: *integer*

RAS Event number.

value	symbol
1	RASEV_OFFLINE
2	RASEV_RTE
3	RASEV_CODETAMPER
4	RASEV_TAMPER
5	RASEV_DURESS
6	RASEV_CARD
7	RASEV_PIN
8	RASEV_DOORACC
9	RASEV_LOCKED
10	RASEV_ISOLATE
11	RASEV_DOORBELL
12	RASEV_CARDV
13	RASEV_EXIT_START
14	RASEV_ENTRY_STOPPED

event1.dgp

multiplicity: *single (static)*
type: *integer*

DGP Event number.

value	symbol
1	DGPEV_OFFLINE
2	DGPEV_MAINSFAIL
3	DGPEV_BATTFAIL
4	DGPEV_TAMPER

value	symbol
5	DGPEV_FUSEFAULT
6	DGPEV_SIRENFAULT
7	DGPEV_RCVFAULT
8	DGPEV_ISOLATE
9	DGPEV_BATTLOW
10	DGPEV_BTESTACTV
11	DGPEV_BTESTFAIL
12	DGPEV_PU_FAIL

event1.panel

multiplicity: *single (static)*
type: *integer*

Panel Event number.

value	symbol
1	DGP0EV_MAINSFAIL
2	DGP0EV_BATTFAIL
3	DGP0EV_TAMPER
4	DGP0EV_FUSEFAULT
5	DGP0EV_SIRENFAULT
6	DGP0EV_LF
7	DGP0EV_LFPSTN
8	DGP0EV_LFISDN
9	DGP0EV_LFGSM
10	DGP0EV_FTC
11	DGP0EV_MIFAULT
12	DGP0EV_MIFISDN
13	DGP0EV_MIFGSM
14	DGP0EV_MIFVOICE
15	DGP0EV_NTPF
16	DGP0EV_LFETH
17	DGP0EV_LFIP
18	DGP0EV_LFGPRS
19	DGP0EV_LFIPGPRS
20	DGP0EV_LFTDA
21	DGP0EV_LFTDAGPRS
22	DGP0EV_LFTDAETH
23	DGP0EV_MIFTDA

event1.user

multiplicity: *single (static)*
type: *integer*

User Event number.

value	symbol
1	USREV_CARDPIN
2	USREV_SMSCTRLACTIVE
3	USREV_SMSCTRLLOCK
4	USREV_SMSREPACTIVE
5	USREV_SMSREPAFTERSET

event1.output

multiplicity: *single (static)*
type: *integer*

Output Event number.

value	symbol
1	OUTEV_ACTIVE
2	OUTEV_ON

event1.filter

multiplicity: *single (static)*
type: *integer*

Zone Event number.

value	symbol
1	CFLEV_ACTIVE

event1.system

multiplicity: *single (static)*
type: *integer*

System Event number.

value	symbol
1	SYSEV_ALLSET
2	SYSEV_AUTOANS
3	SYSEV_RCONNACTV
4	SYSEV_RCONNFAIL
5	SYSEV_LPRGACTV
6	SYSEV_RPRGACTV
7	SYSEV_TIMECHG
8	SYSEV_SSAVER
9	SYSEV_ISIREN
10	SYSEV_ESIREN
11	SYSEV_STROBE
12	SYSEV_FAULT
13	SYSEV_TAMPER

value	symbol
14	SYSEV_SERVICEIN

event1.trigger

multiplicity: *single (static)*
type: *integer*

Trigger Event number.

value	symbol
1	TRGEV_KEYFOBSW1
2	TRGEV_KEYFOBSW2
3	TRGEV_KEYFOBSW12
4	TRGEV_REMOTEOUT
5	TRGEV_FKEY
6	TRGEV_SCHEDULE
7	TRGEV_FOB

event1.cs

multiplicity: *single (static)*
type: *integer*

Central Station Event number.

value	symbol
1	CSEV_FTC
2	CSEV_HBF
3	CSEV_BUSY

event1.ug

multiplicity: *single (static)*
type: *integer*

User Group Event number.

value	symbol
1	UGEV_CARDPIN

event1.sexc

multiplicity: *single (static)*
type: *integer*

Schedule Exception Event number.

value	symbol
1	EXCPEV_ACTIVE

event1.scal

multiplicity: *single (static)*
type: *integer*

Schedule Calendar Event number.

value	symbol
1	SCALEV_HOUR
2	SCALEV_DAY
3	SCALEV_MON
4	SCALEV_TUE
5	SCALEV_WED
6	SCALEV_THU
7	SCALEV_FRI
8	SCALEV_SAT
9	SCALEV_SUN

event1.fob

multiplicity: *single (static)*
type: *integer*

Fob Event number.

value	symbol
1	FOBEV_LEARNED
2	FOBEV_BUTTON1
3	FOBEV_BUTTON2
4	FOBEV_BUTTON3
5	FOBEV_BUTTON4
6	FOBEV_BUTTON12
7	FOBEV_BUTTON13
8	FOBEV_BUTTON14
9	FOBEV_BUTTON23
10	FOBEV_BUTTON24
11	FOBEV_BUTTON34

event1.camera

multiplicity: *single (static)*
type: *integer*

Fob Event number.

value	symbol
1	CAMEV_PICTURE_CAPTURED
2	CAMEV_EV_LIMIT_EXCEEDED

class1

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

object class.

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

obj_no1

multiplicity: *single (static)*
type: *integer*

Object number.

obj_no1.zone

multiplicity: *single (static)*
type: *integer*

Zone number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no1.area

multiplicity: *single (static)*
type: *integer*

Area number.

min	max
1	8

value	symbol
0	ANY

obj_no1.ras

multiplicity: *single (static)*
type: *integer*

RAS number.

min	max
1	8

value	symbol
0	ANY

obj_no1.dgp

multiplicity: *single (static)*
type: *integer*

DGP number.

min	max
1	7

value	symbol
0	ANY

obj_no1.user

multiplicity: *single (static)*
type: *integer*

User number.

min	max
1	50

value	symbol
0	ANY

obj_no1.output

multiplicity: *single (static)*
type: *integer*

Output number.

min	max
1	200

value	symbol
0	ANY

obj_no1.filter

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Filter number.

min	max
1	64

obj_no1.trigger

multiplicity: *single (static)*
type: *integer*

Trigger number.

min	max
1	255

value	symbol
0	ANY

obj_no1.ug

multiplicity: *single (static)*
type: *integer*

UG number.

min	max
1	16

value	symbol
0	ANY

obj_no1.sexc

multiplicity: *single (static)*
type: *integer*

Schedule exception number.

min	max
1	64

value	symbol
0	ANY

obj_no1.fob

multiplicity: *single (static)*
type: *integer*

Fob number.

min	max
1	112

value	symbol
0	ANY

obj_no1.camera

multiplicity: *single (static)*
type: *integer*

Camera number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no1_invert

multiplicity: *single (static)*
type: *integer*

Invert

value	symbol
0	false
1	true

event2

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Event number.

event2.zone

multiplicity: *single (static)*
type: *integer*

Zone Event number.

Notes:

- `ZNEV_LEARNED` is available since protocol version `018`.
- `ZNEV_PRELEARNED` is available since protocol version `020`.
- `ZNEV_HELDOPEN` and `ZNEV_INVWT_INACTIVE` are available since protocol version `023`.

value	symbol
1	ZNEV_ACTIVE
2	ZNEV_TAMPER
3	ZNEV_AM
4	ZNEV_BATTFAIL
5	ZNEV_FAULT
6	ZNEV_DIRTY
7	ZNEV_SVSHORT
8	ZNEV_SVLONG
9	ZNEV_INHIBIT
10	ZNEV_ISOLATE
11	ZNEV_SOAK
12	ZNEV_SET
13	ZNEV_ALARM
14	ZNEV_LEARNED
15	ZNEV_PRELEARNED
16	ZNEV_HELDOPEN
17	ZNEV_INVWT_INACTIVE

event2.area

multiplicity: *single (static)*
type: *integer*

Area Event number.

value	symbol
1	AREV_FULLSET
2	AREV_PARTSET
3	AREV_UNSET
4	AREV_ALARM
5	AREV_FSALARM
6	AREV_PSALARM
7	AREV_USALARM
8	AREV_FTCALARM
9	AREV_FIREDOOR
10	AREV_FSFIREDOOR
11	AREV_PSFIREDOOR
12	AREV_USFIREDOOR
13	AREV_FTCFIREDOOR
14	AREV_FIRE
15	AREV_FSFIRE
16	AREV_PSFIRE
17	AREV_USFIRE
18	AREV_FTCFIRE
19	AREV_PANIC
20	AREV_FSPANIC
21	AREV_PSPANIC
22	AREV_USPANIC
23	AREV_FTCPANIC
24	AREV_MEDICAL
25	AREV_FSMEDICAL
26	AREV_PSMEDICAL
27	AREV_USMEDICAL
28	AREV_FTCMEDICAL
29	AREV_TECHNICAL
30	AREV_FSTECHNICAL
31	AREV_PSTECHNICAL
32	AREV_USTECHNICAL
33	AREV_FTCTECHNICAL
34	AREV_TAMPER
35	AREV_FSTAMPER
36	AREV_PSTAMPER
37	AREV_USTAMPER
38	AREV_FTCTAMPER
39	AREV_DOORBELL

value	symbol
40	AREV_PSDOORBEL
41	AREV_USDOORBEL
42	AREV_ZNACTIVE
43	AREV_ZNINHIBIT
44	AREV_ZNISOLATE
45	AREV_ZNFAULT
46	AREV_ZNAM
47	AREV_ZNTAMPER
48	AREV_RASTAMPER
49	AREV_RASFAULT
50	AREV_DGPTAMPER
51	AREV_DGPFAULT
52	AREV_DURESS
53	AREV_FSDURESS
54	AREV_PSDURESS
55	AREV_USDURESS
56	AREV_FTCDURESS
57	AREV_CODETAMPER
58	AREV_ENTRY
59	AREV_EXIT
60	AREV_EXITFAULT
61	AREV_RTS
62	AREV_SETOK
63	AREV_SETFAULT
64	AREV_UNSETOK
65	AREV_ALARMACK
66	AREV_FIRERESET
67	AREV_WALK
68	AREV_WALKZNACTV
69	AREV_AALARM
70	AREV_BALARM
71	AREV_ISIREN
72	AREV_ESIREN
73	AREV_STROBE
74	AREV_BUZZER
75	AREV_AMRESET
76	AREV_PARTSET2
77	AREV_WARNING
78	AREV_AUTOARM
79	AREV_HAALARM
80	AREV_HBALARM

event2.ras

multiplicity: *single (static)*
type: *integer*

RAS Event number.

value	symbol
1	RASEV_OFFLINE
2	RASEV_RTE
3	RASEV_CODETAMPER
4	RASEV_TAMPER
5	RASEV_DURESS
6	RASEV_CARD
7	RASEV_PIN
8	RASEV_DOORACC
9	RASEV_LOCKED
10	RASEV_ISOLATE
11	RASEV_DOORBELL
12	RASEV_CARDV
13	RASEV_EXIT_START
14	RASEV_ENTRY_STOPPED

event2.dgp

multiplicity: *single (static)*
type: *integer*

DGP Event number.

value	symbol
1	DGPEV_OFFLINE
2	DGPEV_MAINSFAIL
3	DGPEV_BATTFAIL
4	DGPEV_TAMPER
5	DGPEV_FUSEFAULT
6	DGPEV_SIRENFAULT
7	DGPEV_RCVFAULT
8	DGPEV_ISOLATE
9	DGPEV_BATTLOW
10	DGPEV_BTESTACTV
11	DGPEV_BTESTFAIL
12	DGPEV_PU_FAIL

event2.panel

multiplicity: *single (static)*
type: *integer*

Panel Event number.

value	symbol
1	DGP0EV_MAINSFAIL
2	DGP0EV_BATTFAIL
3	DGP0EV_TAMPER
4	DGP0EV_FUSEFAULT
5	DGP0EV_SIRENFAULT
6	DGP0EV_LF
7	DGP0EV_LFPSTN
8	DGP0EV_LFISDN
9	DGP0EV_LFGSM
10	DGP0EV_FTC
11	DGP0EV_MIFAULT
12	DGP0EV_MIFISDN
13	DGP0EV_MIFGSM
14	DGP0EV_MIFVOICE
15	DGP0EV_NTPF
16	DGP0EV_LFETH
17	DGP0EV_LFIP
18	DGP0EV_LFGPRS
19	DGP0EV_LFIPGPRS
20	DGP0EV_LFTDA
21	DGP0EV_LFTDAGPRS
22	DGP0EV_LFTDAETH
23	DGP0EV_MIFTDA

event2.user

multiplicity: *single (static)*
type: *integer*

User Event number.

value	symbol
1	USREV_CARDPIN
2	USREV_SMSCTRLACTIVE
3	USREV_SMSCTRLLOCK
4	USREV_SMSREPACTIVE
5	USREV_SMSREPAFTERSET

event2.output

multiplicity: *single (static)*
type: *integer*

Output Event number.

value	symbol
1	OUTEV_ACTIVE
2	OUTEV_ON

event2.filter

multiplicity: *single (static)*
type: *integer*

Zone Event number.

value	symbol
1	CFLEV_ACTIVE

event2.system

multiplicity: *single (static)*
type: *integer*

System Event number.

value	symbol
1	SYSEV_ALLSET
2	SYSEV_AUTOANS
3	SYSEV_RCONNACTV
4	SYSEV_RCONNFAIL
5	SYSEV_LPRGACTV
6	SYSEV_RPRGACTV
7	SYSEV_TIMECHG
8	SYSEV_SSAVER
9	SYSEV_ISIREN
10	SYSEV_ESIREN
11	SYSEV_STROBE
12	SYSEV_FAULT
13	SYSEV_TAMPER
14	SYSEV_SERVICEIN

event2.trigger

multiplicity: *single (static)*
type: *integer*

Trigger Event number.

value	symbol
1	TRGEV_KEYFOBSW1
2	TRGEV_KEYFOBSW2
3	TRGEV_KEYFOBSW12
4	TRGEV_REMOTEOUT
5	TRGEV_FKEY
6	TRGEV_SCHEDULE
7	TRGEV_FOB

event2.sexc

multiplicity: *single (static)*
type: *integer*

Schedule Exception Event number.

value	symbol
1	EXCPEV_ACTIVE

event2.scal

multiplicity: *single (static)*
type: *integer*

Schedule Calendar Event number.

value	symbol
1	SCALEV_HOUR
2	SCALEV_DAY
3	SCALEV_MON
4	SCALEV_TUE
5	SCALEV_WED
6	SCALEV_THU
7	SCALEV_FRI
8	SCALEV_SAT
9	SCALEV_SUN

event2.fob

multiplicity: *single (static)*
type: *integer*

Fob Event number.

value	symbol
1	FOBEV_LEARNED
2	FOBEV_BUTTON1
3	FOBEV_BUTTON2
4	FOBEV_BUTTON3
5	FOBEV_BUTTON4
6	FOBEV_BUTTON12
7	FOBEV_BUTTON13
8	FOBEV_BUTTON14
9	FOBEV_BUTTON23
10	FOBEV_BUTTON24
11	FOBEV_BUTTON34

event2.camera

multiplicity: *single (static)*
type: *integer*

Camera Event number.

value	symbol
1	CAMEV_PICTURE_CAPTURED
2	CAMEV_EV_LIMIT_EXCEEDED

event2.cs

multiplicity: *single (static)*
type: *integer*

Central Station Event number.

value	symbol
1	CSEV_FTC
2	CSEV_HBF
3	CSEV_BUSY

event2.ug

multiplicity: *single (static)*
type: *integer*

User Group Event number.

value	symbol
1	UGEV_CARDPIN

class2

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

object class.

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

obj_no2

multiplicity: *single (static)*
type: *integer*

Object number.

obj_no2.zone

multiplicity: *single (static)*
type: *integer*

Zone number.

min	max
1	128

min	max
257	368

value	symbol
0	ANY

obj_no2.area

multiplicity: *single (static)*
type: *integer*

Area number.

min	max
1	8

value	symbol
0	ANY

obj_no2.ras

multiplicity: *single (static)*
type: *integer*

RAS number.

min	max
1	8

value	symbol
0	ANY

obj_no2.dgp

multiplicity: *single (static)*
type: *integer*

DGP number.

min	max
1	7

value	symbol
0	ANY

obj_no2.user

multiplicity: *single (static)*
type: *integer*

User number.

min	max
1	50

value	symbol
0	ANY

obj_no2.output

multiplicity: *single (static)*
type: *integer*

Output number.

min	max
1	200

value	symbol
0	ANY

obj_no2.filter

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Filter number.

min	max
1	64

obj_no2.trigger

multiplicity: *single (static)*
type: *integer*

Trigger number.

min	max
1	255

value	symbol
0	ANY

obj_no2.ug

multiplicity: *single (static)*
type: *integer*

UG number.

min	max
1	16

value	symbol
0	ANY

obj_no2.sexc

multiplicity: *single (static)*
type: *integer*

Schedule exception number.

min	max
1	64

value	symbol
0	ANY

obj_no2.fob

multiplicity: *single (static)*
type: *integer*

Fob number.

min	max
1	112

value	symbol
0	ANY

obj_no2.camera

multiplicity: *single (static)*
type: *integer*

Camera number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no2_invert

multiplicity: *single (static)*
type: *integer*

Invert

value	symbol
0	false
1	true

event3

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Event number.

event3.zone

multiplicity: *single (static)*
type: *integer*

Zone Event number.

Notes:

- ZNEV_LEARNED is available since protocol version 018.
- ZNEV_PRELEARNED is available since protocol version 020.
- ZNEV_HELDOPEN and ZNEV_INVWT_INACTIVE are available since protocol version 023.

value	symbol
1	ZNEV_ACTIVE
2	ZNEV_TAMPER
3	ZNEV_AM
4	ZNEV_BATTFAIL
5	ZNEV_FAULT
6	ZNEV_DIRTY
7	ZNEV_SVSHORT
8	ZNEV_SVLONG
9	ZNEV_INHIBIT
10	ZNEV_ISOLATE
11	ZNEV_SOAK
12	ZNEV_SET
13	ZNEV_ALARM
14	ZNEV_LEARNED
15	ZNEV_PRELEARNED
16	ZNEV_HELDOPEN
17	ZNEV_INVWT_INACTIVE

event3.area

multiplicity: *single (static)*
type: *integer*

Area Event number.

value	symbol
1	AREV_FULLSET
2	AREV_PARTSET
3	AREV_UNSET
4	AREV_ALARM
5	AREV_FSALARM
6	AREV_PSALARM
7	AREV_USALARM
8	AREV_FTCALARM
9	AREV_FIREDOOR
10	AREV_FSFIREDOOR
11	AREV_PSFIREDOOR
12	AREV_USFIREDOOR
13	AREV_FTCFIREDOOR
14	AREV_FIRE
15	AREV_FSFIRE
16	AREV_PSFIRE
17	AREV_USFIRE
18	AREV_FTCFIRE
19	AREV_PANIC
20	AREV_FSPANIC
21	AREV_PSPANIC
22	AREV_USPANIC
23	AREV_FTCPANIC
24	AREV_MEDICAL
25	AREV_FSMEDICAL
26	AREV_PSMEDICAL
27	AREV_USMEDICAL
28	AREV_FTCMEDICAL
29	AREV_TECHNICAL
30	AREV_FSTECHNICAL
31	AREV_PSTECHNICAL
32	AREV_USTECHNICAL
33	AREV_FTCTECHNICAL
34	AREV_TAMPER
35	AREV_FSTAMPER
36	AREV_PSTAMPER
37	AREV_USTAMPER
38	AREV_FTCTAMPER
39	AREV_DOORBELL

value	symbol
40	AREV_PSDOORBEL
41	AREV_USDOORBEL
42	AREV_ZNACTIVE
43	AREV_ZNINHIBIT
44	AREV_ZNISOLATE
45	AREV_ZNFAULT
46	AREV_ZNAM
47	AREV_ZNTAMPER
48	AREV_RASTAMPER
49	AREV_RASFAULT
50	AREV_DGPTAMPER
51	AREV_DGPFAULT
52	AREV_DURESS
53	AREV_FSDURESS
54	AREV_PSDURESS
55	AREV_USDURESS
56	AREV_FTCDURESS
57	AREV_CODETAMPER
58	AREV_ENTRY
59	AREV_EXIT
60	AREV_EXITFAULT
61	AREV_RTS
62	AREV_SETOK
63	AREV_SETFAULT
64	AREV_UNSETOK
65	AREV_ALARMACK
66	AREV_FIRERESET
67	AREV_WALK
68	AREV_WALKZNACTV
69	AREV_AALARM
70	AREV_BALARM
71	AREV_ISIREN
72	AREV_ESIREN
73	AREV_STROBE
74	AREV_BUZZER
75	AREV_AMRESET
76	AREV_PARTSET2
77	AREV_WARNING
78	AREV_AUTOARM
79	AREV_HAALARM
80	AREV_HBALARM

event3.ras

multiplicity: *single (static)*
type: *integer*

RAS Event number.

value	symbol
1	RASEV_OFFLINE
2	RASEV_RTE
3	RASEV_CODETAMPER
4	RASEV_TAMPER
5	RASEV_DURESS
6	RASEV_CARD
7	RASEV_PIN
8	RASEV_DOORACC
9	RASEV_LOCKED
10	RASEV_ISOLATE
11	RASEV_DOORBELL
12	RASEV_CARDV
13	RASEV_EXIT_START
14	RASEV_ENTRY_STOPPED

event3.dgp

multiplicity: *single (static)*
type: *integer*

DGP Event number.

value	symbol
1	DGPEV_OFFLINE
2	DGPEV_MAINSFAIL
3	DGPEV_BATTFAIL
4	DGPEV_TAMPER
5	DGPEV_FUSEFAULT
6	DGPEV_SIRENFAULT
7	DGPEV_RCVFAULT
8	DGPEV_ISOLATE
9	DGPEV_BATTLOW
10	DGPEV_BTESTACTV
11	DGPEV_BTESTFAIL
12	DGPEV_PU_FAIL

event3.panel

multiplicity: *single (static)*
type: *integer*

Panel Event number.

value	symbol
1	DGP0EV_MAINSFAIL
2	DGP0EV_BATTFAIL
3	DGP0EV_TAMPER
4	DGP0EV_FUSEFAULT
5	DGP0EV_SIRENFAULT
6	DGP0EV_LF
7	DGP0EV_LFPSTN
8	DGP0EV_LFISDN
9	DGP0EV_LFGSM
10	DGP0EV_FTC
11	DGP0EV_MIFAULT
12	DGP0EV_MIFISDN
13	DGP0EV_MIFGSM
14	DGP0EV_MIFVOICE
15	DGP0EV_NTPF
16	DGP0EV_LFETH
17	DGP0EV_LFIP
18	DGP0EV_LFGPRS
19	DGP0EV_LFIPGPRS
20	DGP0EV_LFTDA
21	DGP0EV_LFTDAGPRS
22	DGP0EV_LFTDAETH
23	DGP0EV_MIFTDA

event3.user

multiplicity: *single (static)*
type: *integer*

User Event number.

value	symbol
1	USREV_CARDPIN
2	USREV_SMSCTRLACTIVE
3	USREV_SMSCTRLLOCK
4	USREV_SMSREPACTIVE
5	USREV_SMSREPAFTERSET

event3.output

multiplicity: *single (static)*
type: *integer*

Output Event number.

value	symbol
1	OUTEV_ACTIVE
2	OUTEV_ON

event3.filter

multiplicity: *single (static)*
type: *integer*

Zone Event number.

value	symbol
1	CFLEV_ACTIVE

event3.system

multiplicity: *single (static)*
type: *integer*

System Event number.

value	symbol
1	SYSEV_ALLSET
2	SYSEV_AUTOANS
3	SYSEV_RCONNACTV
4	SYSEV_RCONNFAIL
5	SYSEV_LPRGACTV
6	SYSEV_RPRGACTV
7	SYSEV_TIMECHG
8	SYSEV_SSAVER
9	SYSEV_ISIREN
10	SYSEV_ESIREN
11	SYSEV_STROBE
12	SYSEV_FAULT
13	SYSEV_TAMPER
14	SYSEV_SERVICEIN

event3.trigger

multiplicity: *single (static)*
type: *integer*

Trigger Event number.

value	symbol
1	TRGEV_KEYFOBSW1
2	TRGEV_KEYFOBSW2
3	TRGEV_KEYFOBSW12
4	TRGEV_REMOTEOUT
5	TRGEV_FKEY
6	TRGEV_SCHEDULE
7	TRGEV_FOB

event3.cs

multiplicity: *single (static)*
type: *integer*

Central Station Event number.

value	symbol
1	CSEV_FTC
2	CSEV_HBF
3	CSEV_BUSY

event3.ug

multiplicity: *single (static)*
type: *integer*

User Group Event number.

value	symbol
1	UGEV_CARDPIN

event3.sexc

multiplicity: *single (static)*
type: *integer*

Schedule Exception Event number.

value	symbol
1	EXCPEV_ACTIVE

event3.scal

multiplicity: *single (static)*
type: *integer*

Schedule Calendar Event number.

value	symbol
1	SCALEV_HOUR
2	SCALEV_DAY
3	SCALEV_MON
4	SCALEV_TUE
5	SCALEV_WED
6	SCALEV_THU
7	SCALEV_FRI
8	SCALEV_SAT
9	SCALEV_SUN

event3.fob

multiplicity: *single (static)*
type: *integer*

Fob Event number.

value	symbol
1	FOBEV_LEARNED
2	FOBEV_BUTTON1
3	FOBEV_BUTTON2
4	FOBEV_BUTTON3
5	FOBEV_BUTTON4
6	FOBEV_BUTTON12
7	FOBEV_BUTTON13
8	FOBEV_BUTTON14
9	FOBEV_BUTTON23
10	FOBEV_BUTTON24
11	FOBEV_BUTTON34

event3.camera

multiplicity: *single (static)*
type: *integer*

Camera Event number.

value	symbol
1	CAMEV_PICTURE_CAPTURED
2	CAMEV_EV_LIMIT_EXCEEDED

class3

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

object class.

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

obj_no3

multiplicity: *single (static)*
type: *integer*

Object number.

obj_no3.zone

multiplicity: *single (static)*
type: *integer*

Zone number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no3.area

multiplicity: *single (static)*
type: *integer*

Area number.

min	max
1	8

value	symbol
0	ANY

obj_no3.ras

multiplicity: *single (static)*
type: *integer*

RAS number.

min	max
1	8

value	symbol
0	ANY

obj_no3.dgp

multiplicity: *single (static)*
type: *integer*

DGP number.

min	max
1	7

value	symbol
0	ANY

obj_no3.user

multiplicity: *single (static)*
type: *integer*

User number.

min	max
1	50

value	symbol
0	ANY

obj_no3.output

multiplicity: *single (static)*
type: *integer*

Output number.

min	max
1	200

value	symbol
0	ANY

obj_no3.filter

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Filter number.

min	max
1	64

obj_no3.trigger

multiplicity: *single (static)*
type: *integer*

Trigger number.

min	max
1	255

value	symbol
0	ANY

obj_no3.ug

multiplicity: *single (static)*
type: *integer*

UG number.

min	max
1	16

value	symbol
0	ANY

obj_no3.sexc

multiplicity: *single (static)*
type: *integer*

Schedule exception number.

min	max
1	64

value	symbol
0	ANY

obj_no3.fob

multiplicity: *single (static)*
type: *integer*

Fob number.

min	max
1	112

value	symbol
0	ANY

obj_no3.camera

multiplicity: *single (static)*
type: *integer*

Camera number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no3_invert

multiplicity: *single (static)*
type: *integer*

Invert

value	symbol
0	false
1	true

event4

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Event number.

event4.zone

multiplicity: *single (static)*
type: *integer*

Zone Event number.

Notes:

- ZNEV_LEARNED is available since protocol version 018.
- ZNEV_PRELEARNED is available since protocol version 020.
- ZNEV_HELDOPEN and ZNEV_INVWT_INACTIVE are available since protocol version 023.

value	symbol
1	ZNEV_ACTIVE
2	ZNEV_TAMPER
3	ZNEV_AM
4	ZNEV_BATTFAIL
5	ZNEV_FAULT
6	ZNEV_DIRTY
7	ZNEV_SVSHORT
8	ZNEV_SVLONG
9	ZNEV_INHIBIT
10	ZNEV_ISOLATE
11	ZNEV_SOAK
12	ZNEV_SET
13	ZNEV_ALARM
14	ZNEV_LEARNED
15	ZNEV_PRELEARNED
16	ZNEV_HELDOPEN
17	ZNEV_INVWT_INACTIVE

event4.area

multiplicity: *single (static)*
type: *integer*

Area Event number.

value	symbol
1	AREV_FULLSET
2	AREV_PARTSET
3	AREV_UNSET
4	AREV_ALARM
5	AREV_FSALARM
6	AREV_PSALARM
7	AREV_USALARM
8	AREV_FTCALARM
9	AREV_FIREDOOR
10	AREV_FSFIREDOOR
11	AREV_PSFIREDOOR
12	AREV_USFIREDOOR
13	AREV_FTCFIREDOOR
14	AREV_FIRE
15	AREV_FSFIRE
16	AREV_PSFIRE
17	AREV_USFIRE
18	AREV_FTCFIRE
19	AREV_PANIC
20	AREV_FSPANIC
21	AREV_PSPANIC
22	AREV_USPANIC
23	AREV_FTCPANIC
24	AREV_MEDICAL
25	AREV_FSMEDICAL
26	AREV_PSMEDICAL
27	AREV_USMEDICAL
28	AREV_FTCMEDICAL
29	AREV_TECHNICAL
30	AREV_FSTECHNICAL
31	AREV_PSTECHNICAL
32	AREV_USTECHNICAL
33	AREV_FTCTECHNICAL
34	AREV_TAMPER
35	AREV_FSTAMPER
36	AREV_PSTAMPER
37	AREV_USTAMPER
38	AREV_FTCTAMPER
39	AREV_DOORBELL

value	symbol
40	AREV_PSDOORBEL
41	AREV_USDOORBEL
42	AREV_ZNACTIVE
43	AREV_ZNINHIBIT
44	AREV_ZNISOLATE
45	AREV_ZNFAULT
46	AREV_ZNAM
47	AREV_ZNTAMPER
48	AREV_RASTAMPER
49	AREV_RASFAULT
50	AREV_DGPTAMPER
51	AREV_DGPFAULT
52	AREV_DURESS
53	AREV_FSDURESS
54	AREV_PSDURESS
55	AREV_USDURESS
56	AREV_FTCDURESS
57	AREV_CODETAMPER
58	AREV_ENTRY
59	AREV_EXIT
60	AREV_EXITFAULT
61	AREV_RTS
62	AREV_SETOK
63	AREV_SETFAULT
64	AREV_UNSETOK
65	AREV_ALARMACK
66	AREV_FIRERESET
67	AREV_WALK
68	AREV_WALKZNACTV
69	AREV_AALARM
70	AREV_BALARM
71	AREV_ISIREN
72	AREV_ESIREN
73	AREV_STROBE
74	AREV_BUZZER
75	AREV_AMRESET
76	AREV_PARTSET2
77	AREV_WARNING
78	AREV_AUTOARM
79	AREV_HAALARM
80	AREV_HBALARM

event4.ras

multiplicity: *single (static)*
type: *integer*

RAS Event number.

value	symbol
1	RASEV_OFFLINE
2	RASEV_RTE
3	RASEV_CODETAMPER
4	RASEV_TAMPER
5	RASEV_DURESS
6	RASEV_CARD
7	RASEV_PIN
8	RASEV_DOORACC
9	RASEV_LOCKED
10	RASEV_ISOLATE
11	RASEV_DOORBELL
12	RASEV_CARDV
13	RASEV_EXIT_START
14	RASEV_ENTRY_STOPPED

event4.dgp

multiplicity: *single (static)*
type: *integer*

DGP Event number.

value	symbol
1	DGPEV_OFFLINE
2	DGPEV_MAINSFAIL
3	DGPEV_BATTFAIL
4	DGPEV_TAMPER
5	DGPEV_FUSEFAULT
6	DGPEV_SIRENFAULT
7	DGPEV_RCVFAULT
8	DGPEV_ISOLATE
9	DGPEV_BATTLOW
10	DGPEV_BTESTACTV
11	DGPEV_BTESTFAIL
12	DGPEV_PU_FAIL

event4.panel

multiplicity: *single (static)*
type: *integer*

Panel Event number.

value	symbol
1	DGP0EV_MAINSFAIL
2	DGP0EV_BATTFAIL
3	DGP0EV_TAMPER
4	DGP0EV_FUSEFAULT
5	DGP0EV_SIRENFAULT
6	DGP0EV_LF
7	DGP0EV_LFPSTN
8	DGP0EV_LFISDN
9	DGP0EV_LFGSM
10	DGP0EV_FTC
11	DGP0EV_MIFAULT
12	DGP0EV_MIFISDN
13	DGP0EV_MIFGSM
14	DGP0EV_MIFVOICE
15	DGP0EV_NTPF
16	DGP0EV_LFETH
17	DGP0EV_LFIP
18	DGP0EV_LFGPRS
19	DGP0EV_LFIPGPRS
20	DGP0EV_LFTDA
21	DGP0EV_LFTDAGPRS
22	DGP0EV_LFTDAETH
23	DGP0EV_MIFTDA

event4.user

multiplicity: *single (static)*
type: *integer*

User Event number.

value	symbol
1	USREV_CARDPIN
2	USREV_SMSCTRLACTIVE
3	USREV_SMSCTRLLOCK
4	USREV_SMSREPACTIVE
5	USREV_SMSREPAFTERSET

event4.output

multiplicity: *single (static)*
type: *integer*

Output Event number.

value	symbol
1	OUTEV_ACTIVE
2	OUTEV_ON

event4.filter

multiplicity: *single (static)*
type: *integer*

Zone Event number.

value	symbol
1	CFLEV_ACTIVE

event4.system

multiplicity: *single (static)*
type: *integer*

System Event number.

value	symbol
1	SYSEV_ALLSET
2	SYSEV_AUTOANS
3	SYSEV_RCONNACTV
4	SYSEV_RCONNFAIL
5	SYSEV_LPRGACTV
6	SYSEV_RPRGACTV
7	SYSEV_TIMECHG
8	SYSEV_SSAVER
9	SYSEV_ISIREN
10	SYSEV_ESIREN
11	SYSEV_STROBE
12	SYSEV_FAULT
13	SYSEV_TAMPER
14	SYSEV_SERVICEIN

event4.trigger

multiplicity: *single (static)*
type: *integer*

Trigger Event number.

value	symbol
1	TRGEV_KEYFOBSW1
2	TRGEV_KEYFOBSW2
3	TRGEV_KEYFOBSW12
4	TRGEV_REMOTEOUT
5	TRGEV_FKEY
6	TRGEV_SCHEDULE
7	TRGEV_FOB

event4.cs

multiplicity: *single (static)*
type: *integer*

Central Station Event number.

value	symbol
1	CSEV_FTC
2	CSEV_HBF
3	CSEV_BUSY

event4.ug

multiplicity: *single (static)*
type: *integer*

User Group Event number.

value	symbol
1	UGEV_CARDPIN

event4.sexc

multiplicity: *single (static)*
type: *integer*

Schedule Exception Event number.

value	symbol
1	EXCPEV_ACTIVE

event4.scal

multiplicity: *single (static)*
type: *integer*

Schedule Calendar Event number.

value	symbol
1	SCALEV_HOUR
2	SCALEV_DAY
3	SCALEV_MON
4	SCALEV_TUE
5	SCALEV_WED
6	SCALEV_THU
7	SCALEV_FRI
8	SCALEV_SAT
9	SCALEV_SUN

event4.fob

multiplicity: *single (static)*
type: *integer*

Fob Event number.

value	symbol
1	FOBEV_LEARNED
2	FOBEV_BUTTON1
3	FOBEV_BUTTON2
4	FOBEV_BUTTON3
5	FOBEV_BUTTON4
6	FOBEV_BUTTON12
7	FOBEV_BUTTON13
8	FOBEV_BUTTON14
9	FOBEV_BUTTON23
10	FOBEV_BUTTON24
11	FOBEV_BUTTON34

event4.camera

multiplicity: *single (static)*
type: *integer*

Camera Event number.

value	symbol
1	CAMEV_PICTURE_CAPTURED
2	CAMEV_EV_LIMIT_EXCEEDED

class4

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

object class.

value	symbol
1	APPOBJ_ZN
2	APPOBJ_AREA
3	APPOBJ_RAS
4	APPOBJ_DGP
5	APPOBJ_DGP0
6	APPOBJ_USER
7	APPOBJ_OUT
8	APPOBJ_CEVFLT
9	APPOBJ_UG
10	APPOBJ_SYS
15	APPOBJ_CS
16	APPOBJ_EV
17	APPOBJ_DL
18	APPOBJ_PCC
19	APPOBJ_MISC
20	APPOBJ_TRIGGER
25	APPOBJ_SCHDL_EXC
27	APPOBJ_SCHDL_CAL
28	APPOBJ_FOB
29	APPOBJ_CAMERA

obj_no4

multiplicity: *single (static)*
type: *integer*

Object number.

obj_no4.zone

multiplicity: *single (static)*
type: *integer*

Zone number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no4.area

multiplicity: *single (static)*
type: *integer*

Area number.

min	max
1	8

value	symbol
0	ANY

obj_no4.ras

multiplicity: *single (static)*
type: *integer*

RAS number.

min	max
1	8

value	symbol
0	ANY

obj_no4.dgp

multiplicity: *single (static)*
type: *integer*

DGP number.

min	max
1	7

value	symbol
0	ANY

obj_no4.user

multiplicity: *single (static)*
type: *integer*

User number.

min	max
1	50

value	symbol
0	ANY

obj_no4.output

multiplicity: *single (static)*
type: *integer*

Output number.

min	max
1	200

value	symbol
0	ANY

obj_no4.filter

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Filter number.

min	max
1	64

obj_no4.trigger

multiplicity: *single (static)*
type: *integer*

Trigger number.

min	max
1	255

value	symbol
0	ANY

obj_no4.ug

multiplicity: *single (static)*
type: *integer*

UG number.

min	max
1	16

value	symbol
0	ANY

obj_no4.sexc

multiplicity: *single (static)*
type: *integer*

Schedule exception number.

min	max
1	64

value	symbol
0	ANY

obj_no4.fob

multiplicity: *single (static)*
type: *integer*

Fob number.

min	max
1	112

value	symbol
0	ANY

obj_no4.camera

multiplicity: *single (static)*
type: *integer*

Camera number.

min	max
1	128
257	368

value	symbol
0	ANY

obj_no4_invert

multiplicity: *single (static)*
type: *integer*

Invert

value	symbol
0	false
1	true

op1

multiplicity: *single (static)*
type: *integer*

operator 1.

value	symbol
0	OR
1	AND
2	XOR

op2

multiplicity: *single (static)*
type: *integer*

operator 2.

value	symbol
0	OR
1	AND
2	XOR

op3

multiplicity: *single (static)*
type: *integer*

operator 3.

value	symbol
0	OR
1	AND
2	XOR

index

multiplicity: *single (static)*
type: *integer*

CEvFilter index.

min	max
1	64

delete.Zone

direction: *output*

This is the outgoing call for "deleteZone" method.

index

multiplicity: *single (static)*
type: *integer*

Zone index.

min	max
1	128
129	256
257	368

delete.RAS

direction: *output*

This is the outgoing call for "deleteRAS" method.

index

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

delete.FOB

direction: *output*

This is the outgoing call for "deleteFOB" method.

index

multiplicity: *single (static)*
type: *integer*

FOB index.

min	max
1	112

delete.DGP

direction: *output*

This is the outgoing call for "deleteDGP" method.

index

multiplicity: *single (static)*
type: *integer*

DGP index.

min	max
1	7

delete.User

direction: *output*

This is the outgoing call for "deleteUser" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

delete.Output

direction: *output*

This is the outgoing call for "deleteOUT" method.

index

multiplicity: *single (static)*
type: *integer*

OUTPUT index.

min	max
1	200

delete.UserGroup

direction: *output*

This is the outgoing call for "deleteUserGroup" method.

index

multiplicity: *single (static)*
type: *integer*

UserGroup index.

min	max
1	16

delete.CEvFilter

direction: *output*

This is the outgoing call for "deleteCEvFilter" method.

index

multiplicity: *single (static)*
type: *integer*

CEvFilter index.

min	max
1	64

deleteM.Zone

direction: *output*

This is the outgoing call for "deleteM.Zone" method.

block

multiplicity: *single (static)*
type: *integer*

block

objectBitmask

multiplicity: *single (static)*
type: *integer*

objectBitmask as integer

objectBitmask.1

multiplicity: *single (static)*
type: *boolean*

objectBitmask 1.

objectBitmask.2

multiplicity: *single (static)*
type: *boolean*

objectBitmask 2.

objectBitmask.3

multiplicity: *single (static)*
type: *boolean*

objectBitmask 3.

objectBitmask.4

multiplicity: *single (static)*
type: *boolean*

objectBitmask 4.

objectBitmask.5

multiplicity: *single (static)*
type: *boolean*

objectBitmask 5.

objectBitmask.6

multiplicity: *single (static)*
type: *boolean*

objectBitmask 6.

objectBitmask.7

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.8

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.9

multiplicity: *single (static)*
type: *boolean*

objectBitmask 9.

objectBitmask.10

multiplicity: *single (static)*
type: *boolean*

objectBitmask 10.

objectBitmask.11

multiplicity: *single (static)*
type: *boolean*

objectBitmask 11.

objectBitmask.12

multiplicity: *single (static)*
type: *boolean*

objectBitmask 12.

objectBitmask.13

multiplicity: *single (static)*
type: *boolean*

objectBitmask 13.

objectBitmask.14

multiplicity: *single (static)*
type: *boolean*

objectBitmask 14.

objectBitmask.15

multiplicity: *single (static)*
type: *boolean*

objectBitmask 15.

objectBitmask.16

multiplicity: *single (static)*
type: *boolean*

objectBitmask 16.

deleteM.RAS

direction: *output*

This is the outgoing call for "deleteM.RAS" method.

block

multiplicity: *single (static)*
type: *integer*

block

objectBitmask

multiplicity: *single (static)*
type: *integer*

objectBitmask as integer

objectBitmask.1

multiplicity: *single (static)*
type: *boolean*

objectBitmask 1.

objectBitmask.2

multiplicity: *single (static)*
type: *boolean*

objectBitmask 2.

objectBitmask.3

multiplicity: *single (static)*
type: *boolean*

objectBitmask 3.

objectBitmask.4

multiplicity: *single (static)*
type: *boolean*

objectBitmask 4.

objectBitmask.5

multiplicity: *single (static)*
type: *boolean*

objectBitmask 5.

objectBitmask.6

multiplicity: *single (static)*
type: *boolean*

objectBitmask 6.

objectBitmask.7

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.8

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.9

multiplicity: *single (static)*
type: *boolean*

objectBitmask 9.

objectBitmask.10

multiplicity: *single (static)*
type: *boolean*

objectBitmask 10.

objectBitmask.11

multiplicity: *single (static)*
type: *boolean*

objectBitmask 11.

objectBitmask.12

multiplicity: *single (static)*
type: *boolean*

objectBitmask 12.

objectBitmask.13

multiplicity: *single (static)*
type: *boolean*

objectBitmask 13.

objectBitmask.14

multiplicity: *single (static)*
type: *boolean*

objectBitmask 14.

objectBitmask.15

multiplicity: *single (static)*
type: *boolean*

objectBitmask 15.

objectBitmask.16

multiplicity: *single (static)*
type: *boolean*

objectBitmask 16.

deleteM.FOB

direction: *output*

This is the outgoing call for "deleteM.FOB" method.

block

multiplicity: *single (static)*
type: *integer*

block

objectBitmask

multiplicity: *single (static)*
type: *integer*

objectBitmask as integer

objectBitmask.1

multiplicity: *single (static)*
type: *boolean*

objectBitmask 1.

objectBitmask.2

multiplicity: *single (static)*
type: *boolean*

objectBitmask 2.

objectBitmask.3

multiplicity: *single (static)*
type: *boolean*

objectBitmask 3.

objectBitmask.4

multiplicity: *single (static)*
type: *boolean*

objectBitmask 4.

objectBitmask.5

multiplicity: *single (static)*
type: *boolean*

objectBitmask 5.

objectBitmask.6

multiplicity: *single (static)*
type: *boolean*

objectBitmask 6.

objectBitmask.7

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.8

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.9

multiplicity: *single (static)*
type: *boolean*

objectBitmask 9.

objectBitmask.10

multiplicity: *single (static)*
type: *boolean*

objectBitmask 10.

objectBitmask.11

multiplicity: *single (static)*
type: *boolean*

objectBitmask 11.

objectBitmask.12

multiplicity: *single (static)*
type: *boolean*

objectBitmask 12.

objectBitmask.13

multiplicity: *single (static)*
type: *boolean*

objectBitmask 13.

objectBitmask.14

multiplicity: *single (static)*
type: *boolean*

objectBitmask 14.

objectBitmask.15

multiplicity: *single (static)*
type: *boolean*

objectBitmask 15.

objectBitmask.16

multiplicity: *single (static)*
type: *boolean*

objectBitmask 16.

deleteM.Camera

direction: *output*

This is the outgoing call for "deleteM.Camera" method.

block

multiplicity: *single (static)*
type: *integer*

block

objectBitmask

multiplicity: *single (static)*
type: *integer*

objectBitmask as integer

objectBitmask.1

multiplicity: *single (static)*
type: *boolean*

objectBitmask 1.

objectBitmask.2

multiplicity: *single (static)*
type: *boolean*

objectBitmask 2.

objectBitmask.3

multiplicity: *single (static)*
type: *boolean*

objectBitmask 3.

objectBitmask.4

multiplicity: *single (static)*
type: *boolean*

objectBitmask 4.

objectBitmask.5

multiplicity: *single (static)*
type: *boolean*

objectBitmask 5.

objectBitmask.6

multiplicity: *single (static)*
type: *boolean*

objectBitmask 6.

objectBitmask.7

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.8

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.9

multiplicity: *single (static)*
type: *boolean*

objectBitmask 9.

objectBitmask.10

multiplicity: *single (static)*
type: *boolean*

objectBitmask 10.

objectBitmask.11

multiplicity: *single (static)*
type: *boolean*

objectBitmask 11.

objectBitmask.12

multiplicity: *single (static)*
type: *boolean*

objectBitmask 12.

objectBitmask.13

multiplicity: *single (static)*
type: *boolean*

objectBitmask 13.

objectBitmask.14

multiplicity: *single (static)*
type: *boolean*

objectBitmask 14.

objectBitmask.15

multiplicity: *single (static)*
type: *boolean*

objectBitmask 15.

objectBitmask.16

multiplicity: *single (static)*
type: *boolean*

objectBitmask 16.

deleteM.DGP

direction: *output*

This is the outgoing call for "deleteM.DGP" method.

block

multiplicity: *single (static)*
type: *integer*

block

objectBitmask

multiplicity: *single (static)*
type: *integer*

objectBitmask as integer

objectBitmask.1

multiplicity: *single (static)*
type: *boolean*

objectBitmask 1.

objectBitmask.2

multiplicity: *single (static)*
type: *boolean*

objectBitmask 2.

objectBitmask.3

multiplicity: *single (static)*
type: *boolean*

objectBitmask 3.

objectBitmask.4

multiplicity: *single (static)*
type: *boolean*

objectBitmask 4.

objectBitmask.5

multiplicity: *single (static)*
type: *boolean*

objectBitmask 5.

objectBitmask.6

multiplicity: *single (static)*
type: *boolean*

objectBitmask 6.

objectBitmask.7

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.8

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.9

multiplicity: *single (static)*
type: *boolean*

objectBitmask 9.

objectBitmask.10

multiplicity: *single (static)*
type: *boolean*

objectBitmask 10.

objectBitmask.11

multiplicity: *single (static)*
type: *boolean*

objectBitmask 11.

objectBitmask.12

multiplicity: *single (static)*
type: *boolean*

objectBitmask 12.

objectBitmask.13

multiplicity: *single (static)*
type: *boolean*

objectBitmask 13.

objectBitmask.14

multiplicity: *single (static)*
type: *boolean*

objectBitmask 14.

objectBitmask.15

multiplicity: *single (static)*
type: *boolean*

objectBitmask 15.

objectBitmask.16

multiplicity: *single (static)*
type: *boolean*

objectBitmask 16.

deleteM.User

direction: *output*

This is the outgoing call for "deleteM.User" method.

block

multiplicity: *single (static)*
type: *integer*

block

objectBitmask

multiplicity: *single (static)*
type: *integer*

objectBitmask as integer

objectBitmask.1

multiplicity: *single (static)*
type: *boolean*

objectBitmask 1.

objectBitmask.2

multiplicity: *single (static)*
type: *boolean*

objectBitmask 2.

objectBitmask.3

multiplicity: *single (static)*
type: *boolean*

objectBitmask 3.

objectBitmask.4

multiplicity: *single (static)*
type: *boolean*

objectBitmask 4.

objectBitmask.5

multiplicity: *single (static)*
type: *boolean*

objectBitmask 5.

objectBitmask.6

multiplicity: *single (static)*
type: *boolean*

objectBitmask 6.

objectBitmask.7

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.8

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.9

multiplicity: *single (static)*
type: *boolean*

objectBitmask 9.

objectBitmask.10

multiplicity: *single (static)*
type: *boolean*

objectBitmask 10.

objectBitmask.11

multiplicity: *single (static)*
type: *boolean*

objectBitmask 11.

objectBitmask.12

multiplicity: *single (static)*
type: *boolean*

objectBitmask 12.

objectBitmask.13

multiplicity: *single (static)*
type: *boolean*

objectBitmask 13.

objectBitmask.14

multiplicity: *single (static)*
type: *boolean*

objectBitmask 14.

objectBitmask.15

multiplicity: *single (static)*
type: *boolean*

objectBitmask 15.

objectBitmask.16

multiplicity: *single (static)*
type: *boolean*

objectBitmask 16.

deleteM.Output

direction: *output*

This is the outgoing call for "deleteM.OUT" method.

block

multiplicity: *single (static)*
type: *integer*

block

objectBitmask

multiplicity: *single (static)*
type: *integer*

objectBitmask as integer

objectBitmask.1

multiplicity: *single (static)*
type: *boolean*

objectBitmask 1.

objectBitmask.2

multiplicity: *single (static)*
type: *boolean*

objectBitmask 2.

objectBitmask.3

multiplicity: *single (static)*
type: *boolean*

objectBitmask 3.

objectBitmask.4

multiplicity: *single (static)*
type: *boolean*

objectBitmask 4.

objectBitmask.5

multiplicity: *single (static)*
type: *boolean*

objectBitmask 5.

objectBitmask.6

multiplicity: *single (static)*
type: *boolean*

objectBitmask 6.

objectBitmask.7

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.8

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.9

multiplicity: *single (static)*
type: *boolean*

objectBitmask 9.

objectBitmask.10

multiplicity: *single (static)*
type: *boolean*

objectBitmask 10.

objectBitmask.11

multiplicity: *single (static)*
type: *boolean*

objectBitmask 11.

objectBitmask.12

multiplicity: *single (static)*
type: *boolean*

objectBitmask 12.

objectBitmask.13

multiplicity: *single (static)*
type: *boolean*

objectBitmask 13.

objectBitmask.14

multiplicity: *single (static)*
type: *boolean*

objectBitmask 14.

objectBitmask.15

multiplicity: *single (static)*
type: *boolean*

objectBitmask 15.

objectBitmask.16

multiplicity: *single (static)*
type: *boolean*

objectBitmask 16.

deleteM.UserGroup

direction: *output*

This is the outgoing call for "deleteM.UserGroup" method.

block

multiplicity: *single (static)*
type: *integer*

block

objectBitmask

multiplicity: *single (static)*
type: *integer*

objectBitmask as integer

objectBitmask.1

multiplicity: *single (static)*
type: *boolean*

objectBitmask 1.

objectBitmask.2

multiplicity: *single (static)*
type: *boolean*

objectBitmask 2.

objectBitmask.3

multiplicity: *single (static)*
type: *boolean*

objectBitmask 3.

objectBitmask.4

multiplicity: *single (static)*
type: *boolean*

objectBitmask 4.

objectBitmask.5

multiplicity: *single (static)*
type: *boolean*

objectBitmask 5.

objectBitmask.6

multiplicity: *single (static)*
type: *boolean*

objectBitmask 6.

objectBitmask.7

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.8

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.9

multiplicity: *single (static)*
type: *boolean*

objectBitmask 9.

objectBitmask.10

multiplicity: *single (static)*
type: *boolean*

objectBitmask 10.

objectBitmask.11

multiplicity: *single (static)*
type: *boolean*

objectBitmask 11.

objectBitmask.12

multiplicity: *single (static)*
type: *boolean*

objectBitmask 12.

objectBitmask.13

multiplicity: *single (static)*
type: *boolean*

objectBitmask 13.

objectBitmask.14

multiplicity: *single (static)*
type: *boolean*

objectBitmask 14.

objectBitmask.15

multiplicity: *single (static)*
type: *boolean*

objectBitmask 15.

objectBitmask.16

multiplicity: *single (static)*
type: *boolean*

objectBitmask 16.

deleteM.CEvFilter

direction: *output*

This is the outgoing call for "deleteM.CEvFilter" method.

block

multiplicity: *single (static)*
type: *integer*

block

objectBitmask

multiplicity: *single (static)*
type: *integer*

objectBitmask as integer

objectBitmask.1

multiplicity: *single (static)*
type: *boolean*

objectBitmask 1.

objectBitmask.2

multiplicity: *single (static)*
type: *boolean*

objectBitmask 2.

objectBitmask.3

multiplicity: *single (static)*
type: *boolean*

objectBitmask 3.

objectBitmask.4

multiplicity: *single (static)*
type: *boolean*

objectBitmask 4.

objectBitmask.5

multiplicity: *single (static)*
type: *boolean*

objectBitmask 5.

objectBitmask.6

multiplicity: *single (static)*
type: *boolean*

objectBitmask 6.

objectBitmask.7

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.8

multiplicity: *single (static)*
type: *boolean*

objectBitmask 8.

objectBitmask.9

multiplicity: *single (static)*
type: *boolean*

objectBitmask 9.

objectBitmask.10

multiplicity: *single (static)*
type: *boolean*

objectBitmask 10.

objectBitmask.11

multiplicity: *single (static)*
type: *boolean*

objectBitmask 11.

objectBitmask.12

multiplicity: *single (static)*
type: *boolean*

objectBitmask 12.

objectBitmask.13

multiplicity: *single (static)*
type: *boolean*

objectBitmask 13.

objectBitmask.14

multiplicity: *single (static)*
type: *boolean*

objectBitmask 14.

objectBitmask.15

multiplicity: *single (static)*
type: *boolean*

objectBitmask 15.

objectBitmask.16

multiplicity: *single (static)*
type: *boolean*

objectBitmask 16.

getValid.Areas

direction: *output*

Request for `return.validAreas` with valid areas for command and control purposes.

Remarks

- The message is available since protocol version *021*.

getValid.Cameras

direction: *output*

Request for `return.validCameras` with valid cameras for command and control purposes.

Remarks

- The message is available since protocol version *022*.

return.validAreas

direction: *input*

List of valid areas for command and control purposes.

Remarks

- The message is available since protocol version *021*.

See also

- `getValid.Areas`

bitset

multiplicity: *single (static)*
type: *string*

Set of bits, denoting whether the object is *available* for particular purposes.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the object is available (value equal to 1) or not (value equal to 0).

return.validCameras

direction: *input*

List of valid cameras for command and control purposes.

Remarks

- The message is available since protocol version *022*.

See also

- `getValid.Cameras`

bitset

multiplicity: *single (static)*
type: *string*

Set of bits, denoting whether the object is *available* for particular purposes.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

```
byteIndex = (objectIndex - 1) / 8
```

the index of the byte in the read byte array

```
bitIndex = (objectIndex - 1) % 8
```

the index of the bit in the byte above

The extracted value of the bit informs whether the object is available (value equal to 1) or not (value equal to 0).

getAvailM.Zone

direction: *output*

This is the outgoing call for "getAvailM.Zone" method.

getAvailM.RAS

direction: *output*

This is the outgoing call for "getAvailM.RAS" method.

getAvailM.DGP

direction: *output*

This is the outgoing call for "getAvailM.DGP" method.

getAvailM.User

direction: *output*

This is the outgoing call for "getAvailM.User" method.

getAvailM.Output

direction: *output*

This is the outgoing call for "getAvailM.Output" method.

getAvailM.CEvFilter

direction: *output*

This is the outgoing call for "getAvailM.CEvFilter" method.

getAvailM.UserGroup

direction: *output*

This is the outgoing call for "getAvailM.UserGroup" method.

getAvailM.FOB

direction: *output*

This is the outgoing call for "getAvailM.FOB" method.

getAvailM.Camera

direction: *output*

This is the outgoing call for "getAvailM.Camera" method.

blockID.Zone

direction: *output*

This is the outgoing call for "blockID.Zone" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.Area

direction: *output*

This is the outgoing call for "blockID.Area" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.RAS

direction: *output*

This is the outgoing call for "blockID.RAS" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.DGP

direction: *output*

This is the outgoing call for "blockID.DGP" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.User

direction: *output*

This is the outgoing call for "blockID.User" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.Output

direction: *output*

This is the outgoing call for "blockID.Output" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.Trigger

direction: *output*

This is the outgoing call for "blockID.Trigger" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.CEvFilter

direction: *output*

This is the outgoing call for "blockID.CEvFilter" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.UserGroup

direction: *output*

This is the outgoing call for "blockID.UserGroup" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.CS

direction: *output*

This is the outgoing call for "blockID.CS" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.DL

direction: *output*

This is the outgoing call for "blockID.DL" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.SYS

direction: *output*

This is the outgoing call for "blockIDSYS" method.

The message allows only property *level* to be assigned with value of 0 and property *block* to be assigned with value of 1.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.PCC

direction: *output*

This is the outgoing call for "blockIDPCC" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.SiaEvent

direction: *output*

This is the outgoing call for "blockIDPCC" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.SchedAct

direction: *output*

This is the outgoing call for "blockID.SchedAct" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.SchedActLst

direction: *output*

This is the outgoing call for "blockID.SchedActLst" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.SchedExc

direction: *output*

This is the outgoing call for "blockID.SchedExc" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.Schedule

direction: *output*

This is the outgoing call for "blockID.Schedule" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.Fob

direction: *output*

This is the outgoing call for "blockID.Fob" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.Camera

direction: *output*

This is the outgoing call for "blockID.Camera" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.RemoteUser

direction: *output*

This is the outgoing call for "blockID.RemoteUser" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockID.Master

direction: *output*

This is the outgoing call for "blockID.Master" method.

blockIDM.Zone

direction: *output*

This is the outgoing call for "blockIDM.Zone" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.Area

direction: *output*

This is the outgoing call for "blockIDM.Area" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.RAS

direction: *output*

This is the outgoing call for "blockIDM.RAS" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.DGP

direction: *output*

This is the outgoing call for "blockIDM.DGP" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.User

direction: *output*

This is the outgoing call for "blockIDM.User" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.Output

direction: *output*

This is the outgoing call for "blockIDM.Output" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.Trigger

direction: *output*

This is the outgoing call for "blockIDM.Trigger" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.CEvFilter

direction: *output*

This is the outgoing call for "blockIDM.CEvFilter" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.UserGroup

direction: *output*

This is the outgoing call for "blockIDM.UserGroup" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.CS

direction: *output*

This is the outgoing call for "blockIDM.CS" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.DL

direction: *output*

This is the outgoing call for "blockIDM.DL" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.SYS

direction: *output*

This is the outgoing call for "blockIDM.SYS" method.

The message allows only property *level* to be assigned with value of 0 and property *block* to be assigned with value of 1.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.PCC

direction: *output*

This is the outgoing call for "blockIDM.PCC" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.SiaEvent

direction: *output*

This is the outgoing call for "blockIDM.SiaEvent" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.SchedAct

direction: *output*

This is the outgoing call for "blockIDM.SchedAct" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.SchedActLst

direction: *output*

This is the outgoing call for "blockIDM.SchedActLst" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.SchedExc

direction: *output*

This is the outgoing call for "blockIDM.SchedExc" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.Schedule

direction: *output*

This is the outgoing call for "blockIDM.Schedule" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.Fob

direction: *output*

This is the outgoing call for "blockIDM.Fob" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.Camera

direction: *output*

This is the outgoing call for "blockIDM.Camera" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM.RemoteUser

direction: *output*

This is the outgoing call for "blockIDM.RemoteUser" method.

level

multiplicity: *single (static)*
type: *integer*

level

block

multiplicity: *single (static)*
type: *integer*

block

return.AvailMZone

direction: *input*

This is the incoming message for any "AvailM" method.

bitset

multiplicity: *single (static)*
type: *string*

Set of bits, denoting whether the object is *available* for particular purposes.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

`byteIndex = (objectIndex - 1) / 8`
the index of the byte in the read byte array

`bitIndex = (objectIndex - 1) % 8`
the index of the bit in the byte above

The extracted value of the bit informs whether the object is available (value equal to 1) or not (value equal to 0).

return.AvailMRAS

direction: *input*

This is the incoming message for any "AvailM" method.

bitset

multiplicity: *single (static)*
type: *string*

Set of bits, denoting whether the object is *available* for particular purposes.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

`byteIndex = (objectIndex - 1) / 8`
the index of the byte in the read byte array

`bitIndex = (objectIndex - 1) % 8`
the index of the bit in the byte above

The extracted value of the bit informs whether the object is available (value equal to 1) or not (value equal to 0).

return.AvailMDGP

direction: *input*

This is the incoming message for any "AvailM" method.

bitset

multiplicity: *single (static)*
type: *string*

Set of bits, denoting whether the object is *available* for particular purposes.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

`byteIndex = (objectIndex - 1) / 8`
the index of the byte in the read byte array

`bitIndex = (objectIndex - 1) % 8`
the index of the bit in the byte above

The extracted value of the bit informs whether the object is available (value equal to 1) or not (value equal to 0).

return.AvailMUser

direction: *input*

This is the incoming message for any "AvailM" method.

bitset

multiplicity: *single (static)*
type: *string*

Set of bits, denoting whether the object is *available* for particular purposes.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

`byteIndex = (objectIndex - 1) / 8`
the index of the byte in the read byte array

`bitIndex = (objectIndex - 1) % 8`
the index of the bit in the byte above

The extracted value of the bit informs whether the object is available (value equal to 1) or not (value equal to 0).

return.AvailMOutput

direction: *input*

This is the incoming message for any "AvailM" method.

bitset

multiplicity: *single (static)*
type: *string*

Set of bits, denoting whether the object is *available* for particular purposes.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

`byteIndex = (objectIndex - 1) / 8`
the index of the byte in the read byte array

`bitIndex = (objectIndex - 1) % 8`
the index of the bit in the byte above

The extracted value of the bit informs whether the object is available (value equal to 1) or not (value equal to 0).

return.AvailMCevFilter

direction: *input*

This is the incoming message for any "AvailM" method.

bitset

multiplicity: *single (static)*
type: *string*

Set of bits, denoting whether the object is *available* for particular purposes.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

`byteIndex = (objectIndex - 1) / 8`
the index of the byte in the read byte array

`bitIndex = (objectIndex - 1) % 8`
the index of the bit in the byte above

The extracted value of the bit informs whether the object is available (value equal to 1) or not (value equal to 0).

return.AvailMUserGroup

direction: *input*

This is the incoming message for any "AvailM" method.

bitset

multiplicity: *single (static)*
type: *string*

Set of bits, denoting whether the object is *available* for particular purposes.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

`byteIndex = (objectIndex - 1) / 8`
the index of the byte in the read byte array

`bitIndex = (objectIndex - 1) % 8`
the index of the bit in the byte above

The extracted value of the bit informs whether the object is available (value equal to 1) or not (value equal to 0).

return.AvailMFob

direction: *input*

This is the incoming message for any "AvailM" method.

bitset

multiplicity: *single (static)*
type: *string*

Set of bits, denoting whether the object is *available* for particular purposes.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

`byteIndex = (objectIndex - 1) / 8`
the index of the byte in the read byte array

`bitIndex = (objectIndex - 1) % 8`
the index of the bit in the byte above

The extracted value of the bit informs whether the object is available (value equal to 1) or not (value equal to 0).

return.AvailMCamera

direction: *input*

This is the incoming message for any "AvailM" method.

bitset

multiplicity: *single (static)*
type: *string*

Set of bits, denoting whether the object is *available* for particular purposes.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

Access to bits

The following assumptions are used:

- The object index value is between 1 and *N*.
- The byte array is index is zero-based.
- The byte consists of 8 bits numbered from 0 to 7, where the bit 0 is the lowest and 7 is the highest.

If `objectIndex` consists of the index of the particular objects then its bit status location can be found with

`byteIndex = (objectIndex - 1) / 8`
the index of the byte in the read byte array

`bitIndex = (objectIndex - 1) % 8`
the index of the bit in the byte above

The extracted value of the bit informs whether the object is available (value equal to 1) or not (value equal to 0).

return.BlockId

direction: *input*

This is the incoming message for any request for simple *synchronization stamp* used during synchronization data with the Management Software.

See also

- insert.panelId
- insert.InitKey

blockID

multiplicity: *single (static)*
type: *integer*

block ID

return.BlockIdMaster

direction: *input*

system

multiplicity: *single (static)*
type: *integer*

area

multiplicity: *single (static)*
type: *integer*

dgp

multiplicity: *single (static)*
type: *integer*

ras

multiplicity: *single (static)*
type: *integer*

zone

multiplicity: *single (static)*
type: *integer*

cev-filter

multiplicity: *single (static)*
type: *integer*

output

multiplicity: *single (static)*
type: *integer*

cs

multiplicity: *single (static)*
type: *integer*

pcc
multiplicity: *single (static)*
type: *integer*

dl
multiplicity: *single (static)*
type: *integer*

sia-event
multiplicity: *single (static)*
type: *integer*

trigger
multiplicity: *single (static)*
type: *integer*

schedule-action
multiplicity: *single (static)*
type: *integer*

schedule-action-list
multiplicity: *single (static)*
type: *integer*

schedule-exception
multiplicity: *single (static)*
type: *integer*

schedule
multiplicity: *single (static)*
type: *integer*

user
multiplicity: *single (static)*
type: *integer*

user-group
multiplicity: *single (static)*
type: *integer*

fob

multiplicity: *single (static)*
type: *integer*

camera

multiplicity: *single (static)*
type: *integer*

remote-user

multiplicity: *single (static)*
type: *integer*

return.BlockIdZoneM

direction: *input*

This is the incoming message for any "blockIDM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdAreaM

direction: *input*

This is the incoming message for any "blockIDM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdRASM

direction: *input*

This is the incoming message for any "blockIDM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdDGPM

direction: *input*

This is the incoming message for any "blockIDM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdUserM

direction: *input*

This is the incoming message for any "blockIDM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdOutputM

direction: *input*

This is the incoming message for "blockIDOutputM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdTriggerM

direction: *input*

This is the incoming message for "BlockIdTriggerM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdCevFilterM

direction: *input*

This is the incoming message for "BlockIdCevFilterM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdUserGroupM

direction: *input*

This is the incoming message for "BlockIdUserGroupM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdCSM

direction: *input*

This is the incoming message for "BlockIdCSM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdDLM

direction: *input*

This is the incoming message for "BlockIdDLM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdSysM

direction: *input*

This is the incoming message for "BlockIdSysM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdPCCM

direction: *input*

This is the incoming message for "BlockIdSysM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdSiaEventM

direction: *input*

This is the incoming message for "BlockIdSysM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdSchedActM

direction: *input*

This is the incoming message for "BlockIdSysM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdSchedActLstM

direction: *input*

This is the incoming message for "BlockIdSysM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdSchedExcM

direction: *input*

This is the incoming message for "BlockIdSysM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdScheduleM

direction: *input*

This is the incoming message for "BlockIdSysM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdFobM

direction: *input*

This is the incoming message for "BlockIdSysM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdCameraM

direction: *input*

This is the incoming message for "BlockIdSysM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

return.BlockIdRemoteUserM

direction: *input*

This is the incoming message for "BlockIdSysM" method.

block

multiplicity: *single (static)*
type: *integer*

block

blockIDM

multiplicity: *multiple (dynamic)*
type: *integer*

block ID Multiple

select.SYS1

direction: *output*

This is the outgoing call for "selectSYS1" method.

return.SYS1

direction: *input*

This is the return message for "selectSYS1" method.

sysEOL

multiplicity: *single (static)*
type: *integer*

SYSM_EOL

value	symbol
1	10k
2	4k7
3	2k2
4	6k8
5	5k6
6	3k74
7	3k3
8	2k
9	1k5
10	1k
11	8k2
12	4k7+2k2
255	NOEOL

sysDualLoop

multiplicity: *single (static)*
type: *integer*

SYSM_INPUTMODE

value	symbol
0	SINGLENO
1	DUALLOOP
2	SINGLENC

sysPanicMode

multiplicity: *single (static)*
type: *integer*

SYSM_PANIC

value	symbol
0	Silent
1	Audible
2	Audible LF

sysIntSirenTime

multiplicity: *single (static)*
type: *integer*

SYSM_ISIRTIME

min	max
0	21600

sysExtSirenTime

multiplicity: *single (static)*
type: *integer*

SYSM_ESIRTIME

min	max
0	21600

sysIntSirenDelay

multiplicity: *single (static)*
type: *integer*

SYSM_ISIRDELAY

min	max
0	21600

sysExtSirenDelay

multiplicity: *single (static)*
type: *integer*

SYSM_ESIRDELAY

min	max
0	21600

sysMainsRepDelay

multiplicity: *single (static)*
type: *integer*

SYSM_MAINSDELAY

min	**max**
0	14400

sysSoakTestTime

multiplicity: *single (static)*
type: *integer*

SYSM_SOAKTIME

min	**max**
0	30

sysSwShuntCount

multiplicity: *single (static)*
type: *integer*

SYSM_SWSHUNT

min	**max**
2	4

sysDblKnockTime

multiplicity: *single (static)*
type: *integer*

SYSM_DBLKNOCKTIME

min	**max**
0	900

sysDblKnockOpenTime

multiplicity: *single (static)*
type: *integer*

SYSM_DBLKNOCKOPNTIME

min	**max**
0	900

sysCardPinTimeout

multiplicity: *single (static)*
type: *integer*

SYSM_CARDPINTOUT

min	max
0	300

sysWalktestTimeout

multiplicity: *single (static)*
type: *integer*

SYSM_WLKTESTTOUT

min	max
0	1800

sysScreenSvrTimeout

multiplicity: *single (static)*
type: *integer*

SYSM_SSVRTOUT

min	max
0	300

sysInputActvDelay

multiplicity: *single (static)*
type: *integer*

SYSM_INPACTVDELAY

min	max
0	255

sysAccessToEE

multiplicity: *single (static)*
type: *integer*

SYSM_ACCTOEE

value	symbol
0	false
1	true

sysReportBA

multiplicity: *single (static)*
type: *integer*

SYSM_REPORTBA

value	symbol
0	false
1	true

sysDBPARTSET

multiplicity: *single (static)*
type: *integer*

SYSM_DBPARTSET

value	symbol
0	false
1	true

sysDBAUTO

multiplicity: *single (static)*
type: *integer*

SYSM_DBAUTO

value	symbol
0	false
1	true

sysName

multiplicity: *single (static)*
type: *string*

System name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

insert.SYS1

direction: *output*

This is the outgoing message for "insertSYS1" method.

sysEOL

multiplicity: *single (static)*
type: *integer*

SYSM_EOL

value	symbol
1	10k
2	4k7
3	2k2
4	6k8
5	5k6
6	3k74
7	3k3
8	2k
9	1k5
10	1k
11	8k2
12	4k7+2k2
255	NOEOL

sysDualLoop

multiplicity: *single (static)*
type: *integer*

SYSM_INPUTMODE

value	symbol
0	SINGLENO
1	DUALLOOP
2	SINGLENC

sysPanicMode

multiplicity: *single (static)*
type: *integer*

SYSM_PANIC

value	symbol
0	Silent
1	Audible
2	Audible LF

sysIntSirenTime

multiplicity: *single (static)*
type: *integer*

SYSM_ISIRTIME

min	max
0	21600

sysExtSirenTime

multiplicity: *single (static)*
type: *integer*

SYSM_ESIRTIME

min	max
0	21600

sysIntSirenDelay

multiplicity: *single (static)*
type: *integer*

SYSM_ISIRDELAY

min	max
0	21600

sysExtSirenDelay

multiplicity: *single (static)*
type: *integer*

SYSM_ESIRDELAY

min	max
0	21600

sysMainsRepDelay

multiplicity: *single (static)*
type: *integer*

SYSM_MAINSDELAY

min	max
0	14400

sysSoakTestTime

multiplicity: *single (static)*
type: *integer*

SYSM_SOAKTIME

min	max
0	30

sysSwShuntCount

multiplicity: *single (static)*
type: *integer*

SYSM_SWSHUNT

min	max
2	4

sysDblKnockTime

multiplicity: *single (static)*
type: *integer*

SYSM_DBLKNOCKTIME

min	max
0	900

sysDblKnockOpenTime

multiplicity: *single (static)*
type: *integer*

SYSM_DBLKNOCKOPNTIME

min	max
0	900

sysCardPinTimeout

multiplicity: *single (static)*
type: *integer*

SYSM_CARDPINTOUT

min	max
0	300

sysWalktestTimeout

multiplicity: *single (static)*
type: *integer*

SYSM_WLKTESTTOUT

min	max
0	1800

sysScreenSvrTimeout

multiplicity: *single (static)*
type: *integer*

SYSM_SSVRTOUT

min	max
0	300

sysInputActvDelay

multiplicity: *single (static)*
type: *integer*

SYSM_INPACTVDELAY

min	max
0	255

sysAccessToEE

multiplicity: *single (static)*
type: *integer*

SYSM_ACCTOEE

value	symbol
0	false
1	true

sysReportBA

multiplicity: *single (static)*
type: *integer*

SYSM_REPORTBA

value	symbol
0	false
1	true

sysDBPARTSET

multiplicity: *single (static)*
type: *integer*

SYSM_DBPARTSET

value	symbol
0	false
1	true

sysDBAUTO

multiplicity: *single (static)*
type: *integer*

SYSM_DBAUTO

value	symbol
0	false
1	true

sysName

multiplicity: *single (static)*
type: *string*

System name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

select.SYS2

direction: *output*

This is the outgoing call for "selectSYS2" method.

return.SYS2

direction: *input*

This is the return message for "selectSYS2" method.

sysRTSCHK_ALARM

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_ALARM

value	symbol
0	false
1	true

sysRTSCHK_PANIC

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_PANIC

value	symbol
0	false
1	true

sysRTSCHK_AM

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_AM

value	symbol
0	false
1	true

sysRTSCHK_FAULT

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_FAULT

value	symbol
0	false

value	symbol
1	true

sysRTSCHK_TAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_TAMPER

value	symbol
0	false
1	true

sysRTSCHK_INTCONN

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_INTCONN

value	symbol
0	false
1	true

sysRTSCHK_MAINS

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_MAINS

value	symbol
0	false
1	true

sysRTSCHK_BATTERY

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_BATTERY

value	symbol
0	false
1	true

sysRTSCHK_FTC

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_FTC

value	symbol
0	false
1	true

sysRTSCHK_SIREN

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_SIREN

value	symbol
0	false
1	true

sysRTSCHK_TECHNICAL

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_TECHNICAL

value	symbol
0	false
1	true

sysRTSCHK_RAS

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_RAS

value	symbol
0	false
1	true

sysRTSCHK_DGP

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_DGP

value	symbol
0	false
1	true

sysRTSCHK_TPATH

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_TPATH

value	symbol
0	false
1	true

sysRTSCHK_PALARMS

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_PALARMS

value	symbol
0	false
1	true

sysRTSINH_PANIC

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_PANIC

value	symbol
0	false
1	true

sysRTSINH_AM

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_AM

value	symbol
0	false
1	true

sysRTSINH_FAULT

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_FAULT

value	symbol
0	false
1	true

sysRTSINH_TAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_TAMPER

value	symbol
0	false
1	true

sysRTSINH_INTCONN

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_INTCONN

value	symbol
0	false
1	true

sysRTSINH_MAINS

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_MAINS

value	symbol
0	false
1	true

sysRTSINH_BATTERY

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_BATTERY

value	symbol
0	false
1	true

sysRTSINH_FTC

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_FTC

value	symbol
0	false
1	true

sysRTSINH_SIREN

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_SIREN

value	symbol
0	false
1	true

sysRTSINH_TECHNICAL

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_TECHNICAL

value	symbol
0	false
1	true

sysRTSINH_RAS

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_RAS

value	symbol
0	false
1	true

SYSM_RTSINH_DGP

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_DGP

value	symbol
0	false
1	true

sysRTSINH_TPATH

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_TPATH

value	symbol
0	false
1	true

sysFORCEDSETMODE

multiplicity: *single (static)*
type: *integer*

SYSM_FORCEDSETMODE

value	symbol
0	Inh UnSet
1	Inh Exit
2	Inh Close

sysFORCEDSET

multiplicity: *single (static)*
type: *integer*

SYSM_FORCEDSET

value	symbol
0	false
1	true

sysRTSCHK_ENGRESET

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_ENGRESET

value	symbol
0	false
1	true

sysRTSINH_ENGRESET

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_ENGRESET

value	symbol
0	false
1	true

insert.SYS2

direction: *output*

This is the return message for "insertSYS2" method.

sysRTSCHK_ALARM

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_ALARM

value	symbol
0	false
1	true

sysRTSCHK_PANIC

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_PANIC

value	symbol
0	false
1	true

sysRTSCHK_AM

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_AM

value	symbol
0	false
1	true

sysRTSCHK_FAULT

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_FAULT

value	symbol
0	false

value	symbol
1	true

sysRTSCHK_TAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_TAMPER

value	symbol
0	false
1	true

sysRTSCHK_INTCONN

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_INTCONN

value	symbol
0	false
1	true

sysRTSCHK_MAINS

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_MAINS

value	symbol
0	false
1	true

sysRTSCHK_BATTERY

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_BATTERY

value	symbol
0	false
1	true

sysRTSCHK_FTC

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_FTC

value	symbol
0	false
1	true

sysRTSCHK_SIREN

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_SIREN

value	symbol
0	false
1	true

sysRTSCHK_TECHNICAL

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_TECHNICAL

value	symbol
0	false
1	true

sysRTSCHK_RAS

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_RAS

value	symbol
0	false
1	true

sysRTSCHK_DGP

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_DGP

value	symbol
0	false
1	true

sysRTSCHK_TPATH

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_TPATH

value	symbol
0	false
1	true

sysRTSCHK_PALARMS

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_PALARMS

value	symbol
0	false
1	true

sysRTSINH_PANIC

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_PANIC

value	symbol
0	false
1	true

sysRTSINH_AM

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_AM

value	symbol
0	false
1	true

sysRTSINH_FAULT

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_FAULT

value	symbol
0	false
1	true

sysRTSINH_TAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_TAMPER

value	symbol
0	false
1	true

sysRTSINH_INTCONN

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_INTCONN

value	symbol
0	false
1	true

sysRTSINH_MAINS

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_MAINS

value	symbol
0	false
1	true

sysRTSINH_BATTERY

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_BATTERY

value	symbol
0	false
1	true

sysRTSINH_FTC

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_FTC

value	symbol
0	false
1	true

sysRTSINH_SIREN

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_SIREN

value	symbol
0	false
1	true

sysRTSINH_TECHNICAL

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_TECHNICAL

value	symbol
0	false
1	true

sysRTSINH_RAS

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_RAS

value	symbol
0	false
1	true

SYSM_RTSINH_DGP

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_DGP

value	symbol
0	false
1	true

sysRTSINH_TPATH

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_TPATH

value	symbol
0	false
1	true

sysFORCEDSETMODE

multiplicity: *single (static)*
type: *integer*

SYSM_FORCEDSETMODE

value	symbol
0	Inh UnSet
1	Inh Exit
2	Inh Close

sysFORCEDSET

multiplicity: *single (static)*
type: *integer*

SYSM_FORCEDSET

value	symbol
0	false
1	true

sysRTSCHK_ENGRESET

multiplicity: *single (static)*
type: *integer*

SYSM_RTSCHK_ENGRESET

value	symbol
0	false
1	true

sysRTSINH_ENGRESET

multiplicity: *single (static)*
type: *integer*

SYSM_RTSINH_ENGRESET

value	symbol
0	false
1	true

select.SYS3

direction: *output*

This is the outgoing call for "selectSYS3" method.

return.SYS3

direction: *input*

This is the return message for "selectSYS3" method.

sysConfSystem

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_SYSTEM_MSK.

value	symbol
0	false
1	true

sysConfEE

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_EE_MSK.

value	symbol
0	false
1	true

sysRepEventDelay

multiplicity: *single (static)*
type: *integer*

System event delay.

min	max
0	250

sysABTime

multiplicity: *single (static)*
type: *integer*

SYSM_AB_TIME .

min	max
1	60

sysPCCCallBack

multiplicity: *single (static)*
type: *integer*

SYSM_PCC_CALLBACK .

min	max
0	250

SYSM_ENG_TAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_TAMPER .

value	symbol
0	false
1	true

SYSM_ENG_PANIC

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_PANIC .

value	symbol
0	false
1	true

SYSM_ENG_USER_ACCEPT

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_USER_ACCEPT.

value	symbol
0	false
1	true

SYSM_ENG_PROTECT

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_PROTECT.

value	symbol
0	false
1	true

SYSM_WT_REMINDER

multiplicity: *single (static)*
type: *integer*

SYSM_WT_REMINDER.

value	symbol
0	WT_NEVER
1	WT_EVERY_ARM
2	WT_FIRST_ARM_DAY
3	WT_FIRST_ARM_WEEK
4	WT_FIRST_ARM_MONTH

SYSM_WT_TOSIRENS

multiplicity: *single (static)*
type: *integer*

SYSM_WT_TOSIRENS.

value	symbol
0	false
1	true

SYSM_WT_LOGUNTESTED

multiplicity: *single (static)*
type: *integer*

SYSM_WT_LOGUNTESTED.

value	symbol
0	false
1	true

SYSM_WT_INPTAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_WT_INPTAMPER.

value	symbol
0	false
1	true

SYSM_WT_RASDGPTAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_WT_RASDGPTAMPER.

value	symbol
0	false
1	true

SYSM_WT_SIRENTAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_WT_SIRENTAMPER.

value	symbol
0	false
1	true

SYSM_WT_WALKTESTTOARM

multiplicity: *single (static)*
type: *integer*

SYSM_WT_WALKTESTTOARM.

value	symbol
0	false
1	true

sysCustomText

multiplicity: *single (static)*
type: *string*

Custom text.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

SYSM_SCRSVR_MODE

multiplicity: *single (static)*
type: *integer*

Screensaver mode.

value	symbol
0	off
1	If set
2	Always
3	WithoutCode

SYSM_PINMODE

multiplicity: *single (static)*
type: *integer*

SYSM_PINMODE.

value	symbol
0	SYO_PINMODE_RANDOM
1	SYO_PINMODE_CUSTOM

SYSM_DST1_MTH

multiplicity: *single (static)*
type: *integer*

SYSM_DST1_MTH.

value	symbol
0	MTH_JAN
1	MTH_FEB
2	MTH_MAR
3	MTH_APR
4	MTH_MAY

value	symbol
5	MTH_JUN
6	MTH_JUL
7	MTH_AUG
8	MTH_SEP
9	MTH_OCT
10	MTH_NOV
11	MTH_DEC

SYSM_DST2_MTH

multiplicity: *single (static)*
type: *integer*

SYSM_DST2_MTH.

value	symbol
0	MTH_JAN
1	MTH_FEB
2	MTH_MAR
3	MTH_APR
4	MTH_MAY
5	MTH_JUN
6	MTH_JUL
7	MTH_AUG
8	MTH_SEP
9	MTH_OCT
10	MTH_NOV
11	MTH_DEC

SYSM_DST1_MODE

multiplicity: *single (static)*
type: *integer*

SYSM_DST1_MODE.

value	symbol
1	DST_WK1
2	DST_WK2
3	DST_WK3
4	DST_WK4
5	DST_WKLAST

SYSM_DST2_MODE

multiplicity: *single (static)*
type: *integer*

SYSM_DST2_MODE.

value	symbol
1	DST_WK1
2	DST_WK2
3	DST_WK3
4	DST_WK4
5	DST_WKLAST

SYSM_UTC_OFFSET

multiplicity: *single (static)*
type: *integer*

SYSM_UTC_OFFSET.

value	symbol
0	BTZ_UTC_0
1	BTZ_UTC_P01
2	BTZ_UTC_P02
3	BTZ_UTC_P03
4	BTZ_UTC_P03_30
5	BTZ_UTC_P04
6	BTZ_UTC_P05
7	BTZ_UTC_P05_30
8	BTZ_UTC_P06
9	BTZ_UTC_P07
10	BTZ_UTC_P08
11	BTZ_UTC_P09
12	BTZ_UTC_P09_30
13	BTZ_UTC_P10
14	BTZ_UTC_P10_30
15	BTZ_UTC_P11
16	BTZ_UTC_P12
17	BTZ_UTC_M11
18	BTZ_UTC_M10
19	BTZ_UTC_M09
20	BTZ_UTC_M08
21	BTZ_UTC_M07
22	BTZ_UTC_M06
23	BTZ_UTC_M05
24	BTZ_UTC_M04
25	BTZ_UTC_M03_30

value	symbol
26	BTZ_UTC_M03
27	BTZ_UTC_M02_30
28	BTZ_UTC_M02
29	BTZ_UTC_M01

SYSM_PCC_PASS

multiplicity: *single (static)*
type: *string*

SYSM_PCC_PASS.

SYSM_RP_TIME

multiplicity: *single (static)*
type: *integer*

SYSM_RP_TIME.

min	max
0	1439

SYSM_DIS_PENDINGALM

multiplicity: *single (static)*
type: *integer*

SYSM_DIS_PENDINGALM.

value	symbol
0	false
1	true

SYSM_FINALSETDELAY

multiplicity: *single (static)*
type: *integer*

SYSM_FINALSETDELAY.

SYSM_INSTALLERLOCKOUT

multiplicity: *single (static)*
type: *integer*

SYSM_INSTALLERLOCKOUT.

value	symbol
0	false

value	symbol
1	true

SYSM_CONF_TA_MSK

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_TA_MSK.

value	symbol
0	false
1	true

SYSM_CONF_CALL_CS_MSK

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_CALL_CS_MSK.

value	symbol
0	false
1	true

SYSM_ALARMLIST

multiplicity: *single (static)*
type: *integer*

SYSM_ALARMLIST.

Note: For protocol version below `021` the only valid options are *Off* and *On*.

value	symbol
0	Off
1	On
2	Instant

SYSM_KEYBOXTIME

multiplicity: *single (static)*
type: *integer*

SYSM_KEYBOXTIME.

min	max
1	99

SYSM_ENG_ALARM

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_ALARM.

value	symbol
0	false
1	true

SYSM_CONF_ENG_RES_MSK

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_ENG_RES_MSK.

value	symbol
0	Off
1	A-alarm
2	B-alarm

SYSM_RP_PERIOD

multiplicity: *single (static)*
type: *integer*

SYSM_RP_PERIOD.

min	max
0	999

SYSM_ENGRES_SYSCODE

multiplicity: *single (static)*
type: *integer*

SYSM_ENGRES_SYSCODE.

min	max
0	65535

SYSM_CONF_EEINH_MSK

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_EEINH_MSK.

value	symbol
0	false
1	true

SYSM_RP_EXTEND

multiplicity: *single (static)*
type: *integer*

SYSM_RP_EXTEND

value	symbol
0	false
1	true

SYSM_SERVICETIME

multiplicity: *single (static)*
type: *integer*

SYSM_SERVICETIME.

min	max
60	43200

SYSM_TAMPER_AREAS.1

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.1

SYSM_TAMPER_AREAS.2

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.2

SYSM_TAMPER_AREAS.3

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.3

SYSM_TAMPER_AREAS.4

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.4

SYSM_TAMPER_AREAS.5

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.5

SYSM_TAMPER_AREAS.6

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.6

SYSM_TAMPER_AREAS.7

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.7

SYSM_TAMPER_AREAS.8

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.8

SYSM_OPEN_BOX_TIMER

multiplicity: *single (static)*
type: *integer*

SYSM_OPEN_BOX_TIMER

min	**max**
60	1800

SYSM_TIME_CORRECTION_MODE

multiplicity: *single (static)*
type: *integer*

SYSM_TIME_CORRECTION_MODE

value	symbol
0	SYO_TCM_NONE
1	SYO_TCM_MANUAL
2	SYO_TCM_NTP

SYSM_TIME_CORRECTION_FACTOR

multiplicity: *single (static)*
type: *integer*

SYSM_TIME_CORRECTION_FACTOR

min	max
0	340
-340	-1

SYSM_LISTENING_PORT

multiplicity: *single (static)*
type: *integer*

SYSM_LISTENING_PORT

min	max
0	65535

SYSM_EXTSIREN_RETRIGGER

multiplicity: *single (static)*
type: *integer*

SYSM_EXTSIREN_RETRIGGER

value	symbol
0	false
1	true

SYSM_ACTIVE_SCHED

multiplicity: *single (static)*
type: *integer*

SYSM_ACTIVE_SCHED

min	max
0	4

SYSM_AA_RETRY_TIME

multiplicity: *single (static)*
type: *integer*

SYSM_AA_RETRY_TIME

value	symbol
0	AA_TIMER_OFF
1	AA_TIMER_15M
2	AA_TIMER_30M
3	AA_TIMER_1H
4	AA_TIMER_2H
5	AA_TIMER_3H
6	AA_TIMER_4H

SYSM_AA_USER_RETRY_TIME

multiplicity: *single (static)*
type: *integer*

SYSM_AA_USER_RETRY_TIME

value	symbol
0	AA_TIMER_OFF
1	AA_TIMER_15M
2	AA_TIMER_30M
3	AA_TIMER_1H
4	AA_TIMER_2H
5	AA_TIMER_3H
6	AA_TIMER_4H

SYSM_SIREN_TAMPER_EOL

multiplicity: *single (static)*
type: *integer*

SYSM_AA_USER_RETRY_TIME

value	symbol
1	10k
2	4k7
3	2k2
4	6k8
5	5k6
6	3k74
7	3k3
8	2k
9	1k5

value	symbol
10	1k
11	8k2
12	4k7+2k2
255	NOEOL

SYSM_SS_AND_IP_POLL_ON_SET

multiplicity: *single (static)*
type: *integer*

SYSM_SS_AND_IP_POLL_ON_SET

value	symbol
0	false
1	true

SYSM_RES_REP

multiplicity: *single (static)*
type: *integer*

SYSM_RES_REP

value	symbol
0	ONCONFIRM
1	ONCLOSE

SYSM_RAS_READIN

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

User card learn-in RAS

min	max
1	8

SYSM_TEST_INPUTS

multiplicity: *single (static)*
type: *integer*

Inputs test view

value	symbol
0	All
1	IfUsed

SYSM_ENGRESET_AUTO

multiplicity: *single (static)*
type: *integer*

Automatic engineer reset

value	symbol
0	false
1	true

SYSM_ENGRESET_DISABLE

multiplicity: *single (static)*
type: *integer*

Disable engineer reset when engineer on the system

value	symbol
0	false
1	true

SYSM_BUZZER_MODE

multiplicity: *single (static)*
type: *integer*

Buzzer mode

value	symbol
0	Continuous
1	Intermittent

SYSM_HAB_TIME

multiplicity: *single (static)*
type: *integer*

Holdup AB time

min	max
1	1200

SYSM_CONF_30SEC_DELAY

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_30SEC_DELAY.

value	symbol
0	false
1	true

SYSM_ENG_BATTFAIL

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_BATTFAIL.

value	symbol
0	false
1	true

SYSM_ENG_AUXFUSE

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_AUXFUSE.

value	symbol
0	false
1	true

SYSM_ENG_MAINSFAIL

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_MAINSFAIL.

value	symbol
0	false
1	true

SYSM_ENG_SIRENFAULT

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_SIRENFAULT.

value	symbol
0	false
1	true

SYSM_ENG_INTCONNFAULT

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_INTCONNFAULT.

value	symbol
0	false
1	true

insert.SYS3

direction: *output*

This is the outgoing message for "insertSYS3" method.

sysConfSystem

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_SYSTEM_MSK.

value	symbol
0	false
1	true

sysConfEE

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_EE_MSK.

value	symbol
0	false
1	true

sysRepEventDelay

multiplicity: *single (static)*
type: *integer*

System Retry Time .

min	max
0	250

sysABTime

multiplicity: *single (static)*
type: *integer*

SYSM_AB_TIME .

min	max
1	60

sysPCCCallBack

multiplicity: *single (static)*
type: *integer*

SYSM_PCC_CALLBACK .

min	max
0	250

SYSM_ENG_TAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_TAMPER .

value	symbol
0	false
1	true

SYSM_ENG_PANIC

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_PANIC .

value	symbol
0	false
1	true

SYSM_ENG_USER_ACCEPT

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_USER_ACCEPT.

value	symbol
0	false
1	true

SYSM_ENG_PROTECT

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_PROTECT.

value	symbol
0	false
1	true

SYSM_WT_REMINDER

multiplicity: *single (static)*
type: *integer*

SYSM_WT_REMINDER.

value	symbol
0	WT_NEVER
1	WT_EVERY_ARM
2	WT_FIRST_ARM_DAY
3	WT_FIRST_ARM_WEEK
4	WT_FIRST_ARM_MONTH

SYSM_WT_TOSIRENS

multiplicity: *single (static)*
type: *integer*

SYSM_WT_TOSIRENS.

value	symbol
0	false
1	true

SYSM_WT_LOGUNTESTED

multiplicity: *single (static)*
type: *integer*

SYSM_WT_LOGUNTESTED.

value	symbol
0	false
1	true

SYSM_WT_INPTAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_WT_INPTAMPER.

value	symbol
0	false
1	true

SYSM_WT_RASDGPTAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_WT_RASDGPTAMPER.

value	symbol
0	false
1	true

SYSM_WT_SIRENTAMPER

multiplicity: *single (static)*
type: *integer*

SYSM_WT_SIRENTAMPER.

value	symbol
0	false
1	true

SYSM_WT_WALKTESTTOARM

multiplicity: *single (static)*
type: *integer*

SYSM_WT_WALKTESTTOARM.

value	symbol
0	false
1	true

sysCustomText

multiplicity: *single (static)*
type: *string*

Custom text.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

SYSM_SCRSVR_MODE

multiplicity: *single (static)*
type: *integer*

Screensaver mode.

value	symbol
0	off
1	If set
2	Always
3	WithoutCode

SYSM_PINMODE

multiplicity: *single (static)*
type: *integer*

SYSM_PINMODE.

value	symbol
0	SYO_PINMODE_RANDOM
1	SYO_PINMODE_CUSTOM

SYSM_DST1_MTH

multiplicity: *single (static)*
type: *integer*

SYSM_DST1_MTH.

value	symbol
0	MTH_JAN
1	MTH_FEB
2	MTH_MAR
3	MTH_APR
4	MTH_MAY

value	symbol
5	MTH_JUN
6	MTH_JUL
7	MTH_AUG
8	MTH_SEP
9	MTH_OCT
10	MTH_NOV
11	MTH_DEC

SYSM_DST2_MTH

multiplicity: *single (static)*
type: *integer*

SYSM_DST2_MTH.

value	symbol
0	MTH_JAN
1	MTH_FEB
2	MTH_MAR
3	MTH_APR
4	MTH_MAY
5	MTH_JUN
6	MTH_JUL
7	MTH_AUG
8	MTH_SEP
9	MTH_OCT
10	MTH_NOV
11	MTH_DEC

SYSM_DST1_MODE

multiplicity: *single (static)*
type: *integer*

SYSM_DST1_MODE.

value	symbol
1	DST_WK1
2	DST_WK2
3	DST_WK3
4	DST_WK4
5	DST_WKLAST

SYSM_DST2_MODE

multiplicity: *single (static)*
type: *integer*

SYSM_DST2_MODE.

value	symbol
1	DST_WK1
2	DST_WK2
3	DST_WK3
4	DST_WK4
5	DST_WKLAST

SYSM_UTC_OFFSET

multiplicity: *single (static)*
type: *integer*

SYSM_UTC_OFFSET.

value	symbol
0	BTZ_UTC_0
1	BTZ_UTC_P01
2	BTZ_UTC_P02
3	BTZ_UTC_P03
4	BTZ_UTC_P03_30
5	BTZ_UTC_P04
6	BTZ_UTC_P05
7	BTZ_UTC_P05_30
8	BTZ_UTC_P06
9	BTZ_UTC_P07
10	BTZ_UTC_P08
11	BTZ_UTC_P09
12	BTZ_UTC_P09_30
13	BTZ_UTC_P10
14	BTZ_UTC_P10_30
15	BTZ_UTC_P11
16	BTZ_UTC_P12
17	BTZ_UTC_M11
18	BTZ_UTC_M10
19	BTZ_UTC_M09
20	BTZ_UTC_M08
21	BTZ_UTC_M07
22	BTZ_UTC_M06
23	BTZ_UTC_M05
24	BTZ_UTC_M04
25	BTZ_UTC_M03_30

value	symbol
26	BTZ_UTC_M03
27	BTZ_UTC_M02_30
28	BTZ_UTC_M02
29	BTZ_UTC_M01

SYSM_PCC_PASS

multiplicity: *single (static)*
type: *string*

SYSM_PCC_PASS.

SYSM_RP_TIME

multiplicity: *single (static)*
type: *integer*

SYSM_RP_TIME.

min	max
0	1439

SYSM_DIS_PENDINGALM

multiplicity: *single (static)*
type: *integer*

SYSM_DIS_PENDINGALM.

value	symbol
0	false
1	true

SYSM_FINALSETDELAY

multiplicity: *single (static)*
type: *integer*

SYSM_FINALSETDELAY.

SYSM_INSTALLERLOCKOUT

multiplicity: *single (static)*
type: *integer*

SYSM_INSTALLERLOCKOUT.

value	symbol
0	false

value	symbol
1	true

SYSM_CONF_TA_MSK

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_TA_MSK.

value	symbol
0	false
1	true

SYSM_CONF_CALL_CS_MSK

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_CALL_CS_MSK.

value	symbol
0	false
1	true

SYSM_ALARMLIST

multiplicity: *single (static)*
type: *integer*

SYSM_ALARMLIST.

Note: For protocol version below 021 the only valid options are *Off* and *On*.

value	symbol
0	Off
1	On
2	Instant

SYSM_KEYBOXTIME

multiplicity: *single (static)*
type: *integer*

SYSM_KEYBOXTIME.

min	max
1	99

SYSM_ENG_ALARM

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_ALARM.

value	symbol
0	false
1	true

SYSM_CONF_ENG_RES_MSK

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_ENG_RES_MSK.

value	symbol
0	Off
1	A-alarm
2	B-alarm

SYSM_RP_PERIOD

multiplicity: *single (static)*
type: *integer*

SYSM_RP_PERIOD.

min	max
0	999

SYSM_ENGRES_SYSCODE

multiplicity: *single (static)*
type: *integer*

SYSM_ENGRES_SYSCODE.

min	max
0	65535

SYSM_CONF_EEINH_MSK

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_EEINH_MSK.

value	symbol
0	false
1	true

SYSM_RP_EXTEND

multiplicity: *single (static)*
type: *integer*

SYSM_RP_EXTEND

value	symbol
0	false
1	true

SYSM_SERVICETIME

multiplicity: *single (static)*
type: *integer*

SYSM_SERVICETIME.

min	max
60	43200

SYSM_TAMPER_AREAS.1

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.1

SYSM_TAMPER_AREAS.2

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.2

SYSM_TAMPER_AREAS.3

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.3

SYSM_TAMPER_AREAS.4

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.4

SYSM_TAMPER_AREAS.5

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.5

SYSM_TAMPER_AREAS.6

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.6

SYSM_TAMPER_AREAS.7

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.7

SYSM_TAMPER_AREAS.8

multiplicity: *single (static)*
type: *boolean*

SYSM_TAMPER_AREAS.8

SYSM_OPEN_BOX_TIMER

multiplicity: *single (static)*
type: *integer*

SYSM_OPEN_BOX_TIMER

min	**max**
60	1800

SYSM_TIME_CORRECTION_MODE

multiplicity: *single (static)*
type: *integer*

SYSM_TIME_CORRECTION_MODE

value	symbol
0	SYO_TCM_NONE
1	SYO_TCM_MANUAL
2	SYO_TCM_NTP

SYSM_TIME_CORRECTION_FACTOR

multiplicity: *single (static)*
type: *integer*

SYSM_TIME_CORRECTION_FACTOR

min	max
0	340
-340	-1

SYSM_LISTENING_PORT

multiplicity: *single (static)*
type: *integer*

SYSM_LISTENING_PORT

min	max
0	65535

SYSM_EXTSIREN_RETRIGGER

multiplicity: *single (static)*
type: *integer*

SYSM_EXTSIREN_RETRIGGER

value	symbol
0	false
1	true

SYSM_ACTIVE_SCHED

multiplicity: *single (static)*
type: *integer*

SYSM_ACTIVE_SCHED

min	max
0	4

SYSM_AA_RETRY_TIME

multiplicity: *single (static)*
type: *integer*

SYSM_AA_RETRY_TIME

value	symbol
0	AA_TIMER_OFF
1	AA_TIMER_15M
2	AA_TIMER_30M
3	AA_TIMER_1H
4	AA_TIMER_2H
5	AA_TIMER_3H
6	AA_TIMER_4H

SYSM_AA_USER_RETRY_TIME

multiplicity: *single (static)*
type: *integer*

SYSM_AA_USER_RETRY_TIME

value	symbol
0	AA_TIMER_OFF
1	AA_TIMER_15M
2	AA_TIMER_30M
3	AA_TIMER_1H
4	AA_TIMER_2H
5	AA_TIMER_3H
6	AA_TIMER_4H

SYSM_SIREN_TAMPER_EOL

multiplicity: *single (static)*
type: *integer*

SYSM_AA_USER_RETRY_TIME

value	symbol
1	10k
2	4k7
3	2k2
4	6k8
5	5k6
6	3k74
7	3k3
8	2k
9	1k5

value	symbol
10	1k
11	8k2
12	4k7+2k2
255	NOEOL

SYSM_SS_AND_IP_POLL_ON_SET

multiplicity: *single (static)*
type: *integer*

SYSM_SS_AND_IP_POLL_ON_SET

value	symbol
0	false
1	true

SYSM_RES_REP

multiplicity: *single (static)*
type: *integer*

SYSM_RES_REP

value	symbol
0	ONCONFIRM
1	ONCLOSE

SYSM_RAS_READIN

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

User card learn-in RAS

min	max
1	8

SYSM_TEST_INPUTS

multiplicity: *single (static)*
type: *integer*

Inputs test view

value	symbol
0	All
1	IfUsed

SYSM_ENGRESET_AUTO

multiplicity: *single (static)*
type: *integer*

Automatic engineer reset

value	symbol
0	false
1	true

SYSM_ENGRESET_DISABLE

multiplicity: *single (static)*
type: *integer*

Disable engineer reset when engineer on the system

value	symbol
0	false
1	true

SYSM_BUZZER_MODE

multiplicity: *single (static)*
type: *integer*

Buzzer mode

value	symbol
0	Continuous
1	Intermittent

SYSM_HAB_TIME

multiplicity: *single (static)*
type: *integer*

Holdup AB time

min	max
1	1200

SYSM_CONF_30SEC_DELAY

multiplicity: *single (static)*
type: *integer*

SYSM_CONF_30SEC_DELAY.

value	symbol
0	false
1	true

SYSM_ENG_BATTFAIL

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_BATTFAIL.

value	symbol
0	false
1	true

SYSM_ENG_AUXFUSE

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_AUXFUSE.

value	symbol
0	false
1	true

SYSM_ENG_MAINSFAIL

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_MAINSFAIL.

value	symbol
0	false
1	true

SYSM_ENG_SIRENFAULT

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_SIRENFAULT.

value	symbol
0	false
1	true

SYSM_ENG_INTCONNFAULT

multiplicity: *single (static)*
type: *integer*

SYSM_ENG_INTCONNFAULT.

value	symbol
0	false
1	true

select.SYS0

direction: *output*

This is the outgoing call for "selectSYS0" method.

return.SYS0

direction: *input*

This is the return message for "selectSYS0" method.

sysPanelLang

multiplicity: *single (static)*
type: *integer*

SYSM_PANELLANG

value	symbol
208	ENGLISH UK
144	GERMAN
9	DANISH
68	ITALIAN
224	SPANISH
136	FRENCH
64	DUTCH
76	NORWEGIAN-BOKMAL
80	PORTUGUESE
88	SWEDISH
22	POLISH
65	DUTCH BELG
137	FRENCH BELG
25	SLOVAK
13	FINNISH
8	CZECH
28	TURKISH
6	CATALAN
16	HUNGARIAN

sysPanelCountry

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

SYSM_COUNTRY

min	max
0	254

value	symbol
0	DFLT_EN50131
1	DFLT_NONEN
2	DFLT_UK
3	DFLT_DK
4	DFLT_NL
5	DFLT_FR
6	DFLT_IT
7	DFLT_IRL
8	DFLT_PL
9	DFLT_INCERT_FR_BE
10	DFLT_INCERT_NL_BE
11	DFLT_NO
12	DFLT_GE
13	DFLT_NL_EN
14	DFLT_IRL_EN
15	DFLT_DK_G4S
16	DFLT_FR_NFA2P
17	DFLT_SK
18	DFLT_PT
19	DFLT_PT_EN
20	DFLT_SP_EN
21	DFLT_SW
22	DFLT_CZ
23	DFLT_FI
24	DFLT_SP
25	DFLT_SW_LARM
26	DFLT_FR_D2S

sysAuthorization

multiplicity: *single (static)*
type: *integer*

System Authorization method.

value	symbol
0	PIN only
1	Card and PIN

sysDuress

multiplicity: *single (static)*
type: *integer*

System Duress method.

value	symbol
0	Disabled
1	Inc.last digit
2	Add last digit
3	Add first digit

sysPinLen

multiplicity: *single (static)*
type: *integer*

System Pin length .

min	max
4	10

SYSM_ENG_POTD_ACTIVE

multiplicity: *single (static)*
type: *integer*

Flag indicating whether PIN of The Day is active.

value	symbol
0	false
1	true

SYSM_REMOTE_CONFIG

multiplicity: *single (static)*
type: *integer*

Option to enable remote access to the user profile modification.

value	symbol
0	false
1	true

select.SYS4

direction: *output*

This is the outgoing call for "selectSYS4" method.

return.SYS4

direction: *input*

This is the return message for "selectSYS4" method.

VIEW_EE_COUNTER

multiplicity: *single (static)*
type: *integer*

Controls whether to display how much time is left to enter a code to unset the system during entry time, or how much time is left to leave the area which is setting.

value	symbol
0	false
1	true

HELD_OPEN_TIME

multiplicity: *single (static)*
type: *integer*

System wide held open time to indicate zone opened too long status.

min	max
0	64800

INV_WT_TIME

multiplicity: *single (static)*
type: *integer*

The time in days to enable indication whether the particular zone was not triggered during that time. The value 0 disables the functionality.

min	max
0	127

SYSM_WT_TOBUZZERS

multiplicity: *single (static)*
type: *integer*

The option to enable keypads buzzers at zones activations during WalkTest.

value	symbol
0	false
1	true

SYSM_ENABLE_REMOTE_PIN

multiplicity: *single (static)*
type: *integer*

The option to enable remote user PIN.

value	symbol
0	false
1	true

SYSM_INSPECTION_TEXT

multiplicity: *single (static)*
type: *string*

Text.

SYSM_INSPECTION_TIME

multiplicity: *single (static)*
type: *datetime*

SYSM_INSPECTION_TIME

format: `date`

SYSM_EASY_UNSET

multiplicity: *single (static)*
type: *integer*

The option to enable user code entry without confirm.

value	symbol
0	false
1	true

SYSM_INDICATE_FAULTS_ON_SET

multiplicity: *single (static)*
type: *integer*

The option to enable fault indication during setting only. Valid options: *Always* and *On set*

value	symbol
0	false
1	true

insert.SYS4

direction: *output*

This is the outgoing message for "insertSYS4" method.

VIEW_EE_COUNTER

multiplicity: *single (static)*
type: *integer*

Controls whether to display how much time is left to enter a code to unset the system during entry time, or how much time is left to leave the area which is setting.

value	symbol
0	false
1	true

HELD_OPEN_TIME

multiplicity: *single (static)*
type: *integer*

System wide held open time to indicate zone opened too long status.

min	max
0	64800

INV_WT_TIME

multiplicity: *single (static)*
type: *integer*

The time in days to enable indication whether the particular zone was not triggered during that time. The value 0 disables the functionality.

min	max
0	127

SYSM_WT_TOBUZZERS

multiplicity: *single (static)*
type: *integer*

The option to enable keypads buzzers at zones activations during WalkTest.

value	symbol
0	false
1	true

SYSM_ENABLE_REMOTE_PIN

multiplicity: *single (static)*
type: *integer*

The option to enable remote user PIN.

value	symbol
0	false
1	true

SYSM_INSPECTION_TEXT

multiplicity: *single (static)*
type: *string*

Text.

SYSM_INSPECTION_TIME

multiplicity: *single (static)*
type: *datetime*

SYSM_INSPECTION_TIME

format: `date`

SYSM_EASY_UNSET

multiplicity: *single (static)*
type: *integer*

The option to enable user code entry without confirm.

value	symbol
0	false
1	true

SYSM_INDICATE_FAULTS_ON_SET

multiplicity: *single (static)*
type: *integer*

The option to enable fault indication during setting only. Valid options: *Always* and *On set*

value	symbol
0	false
1	true

device.SecondPIN

direction: *output*

This is the outgoing call for "device.getConnect" method. Remote user login

userPIN

multiplicity: *single (static)*
type: *string*

User Pin.

insert.putPIN

direction: *output*

This is the outgoing call for "insert.putPIN" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userPIN

multiplicity: *single (static)*
type: *string*

User Pin.

insert.putRemotePIN

direction: *output*

This is the outgoing call for "insert.putRemotePIN" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userRemotePIN

multiplicity: *single (static)*
type: *string*

User remote Pin.

insert.putCARD

direction: *output*

This is the outgoing call for "insert.putCARD" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userCARD

multiplicity: *single (static)*
type: *string*

User card data.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

select.getPIN

direction: *output*

This is the outgoing call for "select.getPIN" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

select.getRemotePIN

direction: *output*

This is the outgoing call for "select.getRemotePIN" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

select.getCARD

direction: *output*

This is the outgoing call for "select.getCARD" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

return.getPIN

direction: *input*

This is the return message for "select.getPIN", "select.getRemotePIN", "generate.userPIN" and "generate.userRemotePIN" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userPIN

multiplicity: *single (static)*
type: *string*

User pin.

return.getCARD

direction: *input*

This is the return message for "select.getCARD" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userCARD

multiplicity: *single (static)*
type: *string*

User card data.

Note: the only usable data type for this property is `byte[]` and instance describing such type should be used as the `aux` parameter whenever the property need to be read from the message.

return.getRemotePIN

direction: *input*

This is the return message for "select.getRemotePIN" and "generate.userRemotePIN" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userPIN

multiplicity: *single (static)*
type: *string*

User pin.

generate.userPIN

direction: *output*

This is the outgoing call for "generate.userPIN" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

generate.userRemotePIN

direction: *output*

This is the outgoing call for "generate.userRemotePIN" method.

index

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

userID

multiplicity: *single (static)*
type: *integer*

User index.

min	max
1	50

start.Users

direction: *output*

This is the outgoing call for "start.Users" method. After this command ATS1000A will initialize user upload functionality.

stop.Users

direction: *output*

This is the outgoing call for "stop.Users" method. After this command ATS1000A will stop upload functionality.

commit.Users

direction: *output*

This is the outgoing call for "commit.Users" method. After this command ATS1000A will start with new users database.

select.CSAccount

direction: *output*

This is the outgoing call for "select.CSAccount" method.

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

account

multiplicity: *single (static)*
type: *integer*

Account index.

min	max
1	8

return.CSAccount

direction: *input*

This is the return message for "select.CSAccount" method.

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

account

multiplicity: *single (static)*
type: *integer*

Account index.

min	max
1	8

accountNum

multiplicity: *single (static)*
type: *string*

Account number. This string can be 0-6 characters long. Each account number digit is represented by one byte. If account number is shorter then 6 characters then end of account number string is marked by byte 0x00. Account number valid character set is depending on "csPROT" property. If "csPROT" is set to SIA, XSIA or VOICE. Maximum length of (X)SIA / VOICE account number is 6 characters. For SIA, XSIA or Voice the following characters can be used to define account number: 0123456789 (hex: 0x30 - 0x39) ABCDEF(hex: 0x41 - 0x46) If "csPROT" is set to CID. Maximum length of CID account number is 4 characters. For CID the following characters can be used to define account number: 0123456789 (hex: 0x30 - 0x39) BCDEF(hex: 0x42 - 0x46)

return.CSAccount2

direction: *input*

This is the return message for "select.CSAccount" method.

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

account

multiplicity: *single (static)*
type: *integer*

Account index.

min	max
1	8

accountNum

multiplicity: *single (static)*
type: *string*

Account number.

This string can be 0-12 characters long. Each account number digit is represented by one byte. If account number is shorter than 12 characters then end of account number string is marked by byte 0x00.

Account number valid character set is depending on "csPROT" property.

Protocol type	Valid length	Valid characters
SIA, XSIA, VOICE	0 - 6	0123456789 (0x30 - 0x39)
		ABCDEF (0x41 - 0x46)
CID	0 - 4	0123456789 (0x30 - 0x39)
		BCDEF (0x42 - 0x46)

Protocol type	Valid length	Valid characters
VDS	1 - 12	0123456789 (0x30 - 0x39)

insert.CSAccount

direction: *output*

This is the outgoing message for "insert.CSAccount" method.

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

account

multiplicity: *single (static)*
type: *integer*

Account index.

min	max
1	8

accountNum

multiplicity: *single (static)*
type: *string*

Account number. This string can be 0-6 characters long. Each account number digit is represented by one byte. If account number is shorter then 6 characters then end of account number string is marked by byte 0x00. Account number valid character set is depending on "csPROT" property. If "csPROT" is set to SIA, XSIA or VOICE. Maximum length of (X)SIA / VOICE account number is 6 characters. For SIA, XSIA or Voice the following characters can be used to define account number: 0123456789 (hex: 0x30 - 0x39) ABCDEF(hex: 0x41 - 0x46) If "csPROT" is set to CID. Maximum length of CID account number is 4 characters. For CID the following characters can be used to define account number: 0123456789 (hex: 0x30 - 0x39) BCDEF(hex: 0x42 - 0x46)

insert.CSAccount2

direction: *output*

This is the outgoing message for "insert.CSAccount2" method.

index

multiplicity: *single (static)*
type: *integer*

CS index.

min	max
1	16

account

multiplicity: *single (static)*
type: *integer*

Account index.

min	max
1	8

accountNum

multiplicity: *single (static)*
type: *string*

Account number.

This string can be 0-12 characters long. Each account number digit is represented by one byte. If account number is shorter than 12 characters then end of account number string is marked by byte 0x00.

Account number valid character set is depending on "csPROT" property.

Protocol type	Valid length	Valid characters
SIA, XSIA, VOICE	0 - 6	0123456789 (0x30 - 0x39)
		ABCDEF (0x41 - 0x46)
CID	0 - 4	0123456789 (0x30 - 0x39)
		BCDEF (0x42 - 0x46)

Protocol type	Valid length	Valid characters
VDS	1 - 12	0123456789 (0x30 - 0x39)

return.DL_2

direction: *input*

This is the return message for "selectDialer" method.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlDialTone

multiplicity: *single (static)*
type: *integer*

Dialing otption Pulse/DTMF

value	symbol
0	None
1	Default
3	UK
4	Other

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	false
1	true

dlDialing

multiplicity: *single (static)*
type: *integer*

Dialing otption Pulse/DTMF

value	symbol
0	Pulse
1	DTMF

dlEncrypt

multiplicity: *single (static)*
type: *integer*

Encryption ON/OFF.

value	symbol
0	false
1	true

dlMSN

multiplicity: *single (static)*
type: *string*

MSN phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

DLM_USEDHCP

multiplicity: *single (static)*
type: *integer*

Use DHCP YES/NO

value	symbol
0	false
1	true

DLM_AUTODNS

multiplicity: *single (static)*
type: *integer*

Get DNS from DHCP YES/NO

value	symbol
0	false
1	true

DLM_AUTONTP

multiplicity: *single (static)*
type: *integer*

Get NTP from DHCP YES/NO

value	symbol
0	false
1	true

DLM_USEFIREWALL

multiplicity: *single (static)*
type: *integer*

Firewall ON/OFF

value	symbol
0	false
1	true

DLM_REPLYPING

multiplicity: *single (static)*
type: *integer*

Replay on PING ON/OFF

value	symbol
0	false
1	true

DLM_IPADDR

multiplicity: *single (static)*
type: *string*

IP address

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_NETMASK

multiplicity: *single (static)*
type: *string*

Netmask

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_ROUTER

multiplicity: *single (static)*
type: *string*

Gateway

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_DNSSERVER

multiplicity: *single (static)*
type: *string*

DNS server

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_NTPSERVER

multiplicity: *single (static)*
type: *string*

DNS server

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_ETHSPEED

multiplicity: *single (static)*
type: *integer*

Ethernet link speed

value	symbol
0	AUTO
1	10MB
2	100MB

insert.DL_2

direction: *output*

This is the outgoing message for "insertDialer" method.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlDialTone

multiplicity: *single (static)*
type: *integer*

Dialing otption Pulse/DTMF

value	symbol
0	None
1	Default
3	UK
4	Other

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	false
1	true

dlDialing

multiplicity: *single (static)*
type: *integer*

Dialing otption Pulse/DTMF

value	symbol
0	Pulse
1	DTMF

dlEncrypt

multiplicity: *single (static)*
type: *integer*

Encryption ON/OFF.

value	symbol
0	false
1	true

dlMSN

multiplicity: *single (static)*
type: *string*

MSN phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

DLM_USEDHCP

multiplicity: *single (static)*
type: *integer*

Use DHCP YES/NO

value	symbol
0	false
1	true

DLM_AUTODNS

multiplicity: *single (static)*
type: *integer*

Get DNS from DHCP YES/NO

value	symbol
0	false
1	true

DLM_AUTONTP

multiplicity: *single (static)*
type: *integer*

Get NTP from DHCP YES/NO

value	symbol
0	false
1	true

DLM_USEFIREWALL

multiplicity: *single (static)*
type: *integer*

Firewall ON/OFF

value	symbol
0	false
1	true

DLM_REPLYPING

multiplicity: *single (static)*
type: *integer*

Replay on PING ON/OFF

value	symbol
0	false
1	true

DLM_IPADDR

multiplicity: *single (static)*
type: *string*

IP address

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_NETMASK

multiplicity: *single (static)*
type: *string*

Netmask

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_ROUTER

multiplicity: *single (static)*
type: *string*

Gateway

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_DNSSERVER

multiplicity: *single (static)*
type: *string*

DNS server

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_NTPSERVER

multiplicity: *single (static)*
type: *string*

DNS server

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_ETHSPEED

multiplicity: *single (static)*
type: *integer*

Ethernet link speed

value	symbol
0	AUTO
1	10MB
2	100MB

insert.DL_PSTN

direction: *output*

This is the outgoing message for "insertDialer" method, for PSTN dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

dlEncrypt

multiplicity: *single (static)*
type: *integer*

Encryption ON/OFF.

value	symbol
0	false
1	true

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

dlDialTone

multiplicity: *single (static)*
type: *integer*

Dialing option Pulse/DTMF

value	symbol
0	None
1	Default
3	UK
4	Other

dlDialing

multiplicity: *single (static)*
type: *integer*

Dialing otption Pulse/DTMF

value	symbol
0	Pulse
1	DTMF

dlLFDelay

multiplicity: *single (static)*
type: *integer*

PSTN Line Fault detection delay.

min	max
0	255

insert.DL_ISDN

direction: *output*

This is the outgoing message for "insertDialer" method, for ISDN dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

dlEncrypt

multiplicity: *single (static)*
type: *integer*

Encryption ON/OFF.

value	symbol
0	false
1	true

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

dlMSN

multiplicity: *single (static)*
type: *string*

MSN phone number. This phone number string can be 0-20 characters long. Each phone number digit/character is represented by one byte. If phone number is shorter then 20 characters then end of phone number string is marked by byte 0x00. Valid characters: 0123456789 (hex: 0x30 - 0x39) T P (hex: 0x54,0x50)

dlPTP

multiplicity: *single (static)*
type: *integer*

If true - the Point to Point mode is selected. If false - the Point to Multipoint mode is selected.

value	symbol
0	false
1	true

insert.DL_GSM

direction: *output*

This is the outgoing message for "insertDialer" method, for GSM dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

dlEncrypt

multiplicity: *single (static)*
type: *integer*

Encryption ON/OFF.

value	symbol
0	false
1	true

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

DLM_GSMNET

multiplicity: *single (static)*
type: *string*

GSM network code - empty or 5-6 digits

DLM_PINCODE

multiplicity: *single (static)*
type: *string*

SIM card PIN code: 4 digits

DLM_SMSCNUM

multiplicity: *single (static)*
type: *string*

SMS center phone number. Empty, or GSM number starting with '+'

DLM_CREDITCODE

multiplicity: *single (static)*
type: *string*

Credit check code

DLM_CREDITPERIOD

multiplicity: *single (static)*
type: *integer*

Credit check period in days. If 0 - automatic credit check disabled.

min	max
0	99

DLM_CREDITTIME

multiplicity: *single (static)*
type: *integer*

Credit check time (minutes since 00:00)

min	max
0	1439

DLM_MAXSMSMSGS

multiplicity: *single (static)*
type: *integer*

Max. number of SMS reports during 24hours.

Also: Max. number of SMS messages from unknown sources, forwarded to supervisor during 24hours.

If 0 - no check is performed (unlimited reports/forwards).

min	max
0	200

DLM_SMSHEADER

multiplicity: *single (static)*
type: *string*

SMS report header text.

DLM_SMSFORWARD

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

Index of the user to forward unknown SMS messages and automatic credit checks. If 0 - forwarding disabled.

The user selected must be installer, and must belong to the group with SMS Control enabled.

min	max
1	50

DLM_GSMNETMODE

multiplicity: *single (static)*
type: *integer*

If true - only the selected network may be connected by GSM module.

value	symbol
0	false
1	true

DLM_SMSPINREQ

multiplicity: *single (static)*
type: *integer*

If true - user PIN is required at the start of SMS command message.

value	symbol
0	false
1	true

DLM_SMSEXTCHARSET

multiplicity: *single (static)*
type: *integer*

If true - extended character set (UTF16) is allowed in SMS reports and command responses.

value	symbol
0	false
1	true

DLM_CREDITMODE

multiplicity: *single (static)*
type: *integer*

Method of credit check.

value	symbol
0	NONE
1	CODE
2	SMS

DLM_CREDITNUM

multiplicity: *single (static)*
type: *string*

Phone number for SMS-mode credit check. Empty, or GSM number starting with '+'

insert.DL_IP

direction: *output*

This is the outgoing message for "insertDialer" method, for IP dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

dlEncrypt

multiplicity: *single (static)*
type: *integer*

Encryption ON/OFF.

value	symbol
0	false
1	true

dlRingCnt

multiplicity: *single (static)*
type: *integer*

Ring counter. DLM_RINGCNT Range : 1 - 16 (16 infinity)

min	max
1	15

value	symbol
16	infinity

DLM_USEDHCP

multiplicity: *single (static)*
type: *integer*

Use DHCP YES/NO

value	symbol
0	false
1	true

DLM_AUTODNS

multiplicity: *single (static)*
type: *integer*

Get DNS from DHCP YES/NO

value	symbol
0	false
1	true

DLM_AUTONTP

multiplicity: *single (static)*
type: *integer*

Get NTP from DHCP YES/NO

value	symbol
0	false
1	true

DLM_USEFIREWALL

multiplicity: *single (static)*
type: *integer*

Firewall ON/OFF

value	symbol
0	false
1	true

DLM_REPLYPING

multiplicity: *single (static)*
type: *integer*

Replay on PING ON/OFF

value	symbol
0	false
1	true

DLM_IPADDR

multiplicity: *single (static)*
type: *string*

IP address

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_NETMASK

multiplicity: *single (static)*
type: *string*

Netmask

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_ROUTER

multiplicity: *single (static)*
type: *string*

Gateway

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_DNSSERVER

multiplicity: *single (static)*
type: *string*

DNS server

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_NTPSERVER

multiplicity: *single (static)*
type: *string*

DNS server

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_ETHSPEED

multiplicity: *single (static)*
type: *integer*

Ethernet link speed

value	symbol
0	AUTO
1	10MB
2	100MB

DLM_MAXETHPICMSGS

multiplicity: *single (static)*
type: *integer*

Limit for reported pictures per 24h and arm-disarm cycle

min	max
0	200

insert.DL_STEL

direction: *output*

This is the outgoing message for "insertDialer" method, for Safetel dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
`0x20` … `0x7F`	Characters which are possible to display on ATS1000A RAS devices in all supported languages
`0xA0` … `0xFF`	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

insert.DL_CHIRON

direction: *output*

This is the outgoing message for "insertDialer" method, for Chiron dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

insert.DL_75XX

direction: *output*

This is the outgoing message for "insertDialer" method, for 75XX dialer.

index

multiplicity: *single (static)*
type: *integer*

Dialer index.

min	max
1	7

name

multiplicity: *single (static)*
type: *string*

Dialer name.

- This string must be 1-16 characters length.
- An empty string, that is its length is 0, will not be accepted by ATS1000A and fault message will be send as a response.
- If string is shorter then 16 characters end of text string is marked by first occurence of byte 0x00.

Each character is represented by one byte.

Valid character bytes are in the following ranges:

Range	Description
0x20 … 0x7F	Characters which are possible to display on ATS1000A RAS devices in all supported languages
0xA0 … 0xFF	National characters to display on ATS1000A RAS devices that differs depending on code page of the active language.

To properly display national characters on ATS1000A RAS devices *userLanguage* property (user language) and *sysPanelLang* property (panel language) must be set to particular language.

dlType

multiplicity: *single (static)*
type: *integer*

Dialer type.

value	symbol
0	PSTN
1	ISDN-B
2	GSM
3	IP
4	STEL
5	CHIRON
6	ATS75XX

dlLF

multiplicity: *single (static)*
type: *integer*

Line fault monitor ON/OFF.

value	symbol
0	No
1	Yes
2	If used

insertV.DL_GSM_IP

direction: *output*

This is the outgoing message for "insertDialer" method, for GSM dialer, IP part.

DLM_APNNAME

multiplicity: *single (static)*
type: *string*

Access Point Name

A computer protocol that allows panel to access the Internet using the mobile phone network.

DLM_USERNAME

multiplicity: *single (static)*
type: *string*

Specific user name defined by GPRS provider.

DLM_USERPASS

multiplicity: *single (static)*
type: *string*

Specific user password defined by GPRS provider.

DLM_PPPTIMEOUT

multiplicity: *single (static)*
type: *integer*

Timeout value 5 min - 23 h : 59 min, value 23 h : 59 min means permanent connection.

min	max
5	1439

DLM_AUTODNS

multiplicity: *single (static)*
type: *integer*

Get DNS from DHCP YES/NO.

value	symbol
0	false
1	true

DLM_USEFIREWALL

multiplicity: *single (static)*
type: *integer*

Firewall ON/OFF

value	symbol
0	false
1	true

DLM_REPLYPING

multiplicity: *single (static)*
type: *integer*

Replay on PING ON/OFF

value	symbol
0	false
1	true

DLM_DNSSERVER

multiplicity: *single (static)*
type: *string*

DNS server

format: nnn.nnn.nnn.nnn *(decimal representation of an IP address)*

DLM_PPPLF

multiplicity: *single (static)*
type: *integer*

PPP Line Fault

value	symbol
0	false
1	true

DLM_MAXGPRSPICMSGS

multiplicity: *single (static)*
type: *integer*

Picture limit for GPRS link

min	**max**
0	200

insertV.DL_GSM_MMS

direction: *output*

This is the outgoing message for "insertDialer" method, for GSM dialer, MMS part.

DLM_MMS_APN_NAME

multiplicity: *single (static)*
type: *string*

Access Point Name

A computer protocol that allows panel to send MMSs using the mobile phone network.

DLM_MMS_USER_NAME

multiplicity: *single (static)*
type: *string*

Specific user name defined by GSM provider.

DLM_MMS_USER_PASSWORD

multiplicity: *single (static)*
type: *string*

Specific user password defined by GSM provider.

DLM_MMS_SERVER_NAME

multiplicity: *single (static)*
type: *string*

MMS Central address.

DLM_MMS_PROXYADDR

multiplicity: *single (static)*
type: *string*

Proxy address for MMS Central..

DLM_MMS_PROXYPORT

multiplicity: *single (static)*
type: *integer*

Proxy port for MMS Central.

DLM_MMS_MAXMSGS

multiplicity: *single (static)*
type: *integer*

Picture limit for MMS link

min	**max**
0	200

begin.InitKey

direction: *output*

This message is used to probe if changing the initial session key is possible.

The panel response for the message is return.void if the change is possible or fault otherwise.

Remarks:

- The message is available since protocol version *009*.
- The procedure of changing initial encryption key is valid only for secured channels (already encrypted using session key, or via USB).

insert.InitKey

direction: *output*

This message is used to change the initial session key.

The panel response for the message is `return.BlockId`.

Remarks:
- The message is available since protocol version *009*.
- The procedure of changing initial encryption key is valid only for secured channels (already encrypted using session key, or via USB).

encryptionMode

multiplicity: *single (static)*
type: *integer*

Type of the encryption.

value	symbol
0	NONE
1	AES-128

key1

multiplicity: *single (static)*
type: *string*

ASCII string of the 1-st part of the new initial encryption key, zero-filled up to 12-byte field (if neccessary).

key2

multiplicity: *single (static)*
type: *string*

ASCII string of the 2-nd part of the new initial encryption key, zero-filled up to 12-byte field (if neccessary).

key3

multiplicity: *single (static)*
type: *string*

ASCII string of the 3-rd part of the new initial encryption key, zero-filled up to 12-byte field (if neccessary). Only for 192-bit, or 256-bit keys

key4

multiplicity: *single (static)*
type: *string*

ASCII string of the 4-th part of the new initial encryption key, zero-filled up to 12-byte field (if neccessary). Only for 256-bit key

select.SchedAct

direction: *output*

This is the outgoing call for "selectSchedAct" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

selectV.SACTAct

direction: *output*

This is the outgoing call for "selectVSACTAct" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

subindex

multiplicity: *single (static)*
type: *integer*

Schedule actions subindex.

value	symbol
1	void

select.SchedActLst

direction: *output*

This is the outgoing call for "selectSchedActLst" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule action lists index.

min	max
1	32

select.SchedExc

direction: *output*

This is the outgoing call for "selectSchedExc" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule exceptions index.

min	max
1	64

select.Schedule

direction: *output*

This is the outgoing call for "selectSchedule" method.

index

multiplicity: *single (static)*
type: *integer*

Schedules index.

min	max
1	4

select.ScheduleDayActions

direction: *output*

This is the outgoing call for "selectScheduleDayActions" method.

index

multiplicity: *single (static)*
type: *integer*

Schedules index.

min	max
1	4

day

multiplicity: *single (static)*
type: *integer*

Schedules day.

value	symbol
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday

select.Schedule2

direction: *output*

This is the outgoing call for "selectSchedule2" method.

index

multiplicity: *single (static)*
type: *integer*

Schedules index.

min	max
1	4

insert.SchedAct

direction: *output*

This is the outgoing call for "insertSchedAct" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

SACTM_NAME

multiplicity: *single (static)*
type: *string*

Text.

SACTM_STARTTIME

multiplicity: *single (static)*
type: *integer*

Text.

min	max
0	1439

SACTM_FILTER

multiplicity: *single (static)*
type: *integer*

Filter.

min	max
0	64

SACTM_ACTIVATION

multiplicity: *single (static)*
type: *integer*

Text.

value	symbol
0	false
1	true

insertV.SACTActSet

direction: *output*

This is the outgoing call for "insertVSACTActSet" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

subindex

multiplicity: *single (static)*
type: *integer*

Schedule actions subindex.

value	symbol
1	void

SETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Set Area 1.

SETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Set Area 2.

SETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Set Area 3.

SETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Set Area 4.

SETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Set Area 5.

SETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Set Area 6.

SETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Set Area 7.

SETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Set Area 8.

SETM_AREAS

multiplicity: *single (static)*
type: *integer*

Set Areas.

insertV.SACTActUnset

direction: *output*

This is the outgoing call for "insertVSACTActUnset" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

subindex

multiplicity: *single (static)*
type: *integer*

Schedule actions subindex.

value	symbol
1	void

UNSETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Unset Area 1.

UNSETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Unset Area 2.

UNSETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Unset Area 3.

UNSETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Unset Area 4.

UNSETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Unset Area 5.

UNSETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Unset Area 6.

UNSETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Unset Area 7.

UNSETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Unset Area 8.

UNSETM_AREAS

multiplicity: *single (static)*
type: *integer*

Unset Areas.

insertV.SACTActTrigger

direction: *output*

This is the outgoing call for "insertVSACTActTrigger" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

subindex

multiplicity: *single (static)*
type: *integer*

Schedule actions subindex.

value	symbol
1	void

TRIGGERM_INDEX

multiplicity: *single (static)*
type: *integer*

Trigger index.

min	max
1	255

TRIGGERM_STATE

multiplicity: *single (static)*
type: *integer*

Trigger state.

value	symbol
0	CLEAR
1	SET
2	TOGGLE

insertV.SACTActDoorbell

direction: *output*

This is the outgoing call for "insertVSACTActDoorbell" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

subindex

multiplicity: *single (static)*
type: *integer*

Schedule actions subindex.

value	symbol
1	void

DOORBELLM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 1.

DOORBELLM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 2.

DOORBELLM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 3.

DOORBELLM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 4.

DOORBELLM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 5.

DOORBELLM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 6.

DOORBELLM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 7.

DOORBELLM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 8.

DOORBELLM_AREAS

multiplicity: *single (static)*
type: *integer*

Doorbell Areas.

DOORBELLM_STATE

multiplicity: *single (static)*
type: *integer*

Doorbell state.

value	symbol
0	CLEAR

value	symbol
1	SET
2	TOGGLE

insertV.SACTActUGMask

direction: *output*

This is the outgoing call for "insertVSACTActUGMask" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

subindex

multiplicity: *single (static)*
type: *integer*

Schedule actions subindex.

value	symbol
1	void

UGMASKM_INDEX

multiplicity: *single (static)*
type: *integer*

User Group index.

min	max
1	16

UGMASKM_FULLSET

multiplicity: *single (static)*
type: *integer*

FULLSET.

value	symbol
0	false
1	true

UGMASKM_PARTSET

multiplicity: *single (static)*
type: *integer*

PARTSET.

value	symbol
0	false
1	true

UGMASKM_UNSET

multiplicity: *single (static)*
type: *integer*

UNSET.

value	symbol
0	false
1	true

UGMASKM_INHIBIT

multiplicity: *single (static)*
type: *integer*

INHIBIT.

value	symbol
0	false
1	true

UGMASKM_ISOLATE

multiplicity: *single (static)*
type: *integer*

ISOLATE.

value	symbol
0	false
1	true

UGMASKM_TIMEDATE

multiplicity: *single (static)*
type: *integer*

TIMEDATE.

value	symbol
0	false
1	true

UGMASKM_FSET

multiplicity: *single (static)*
type: *integer*

FSET.

value	symbol
0	false
1	true

UGMASKM_CHANGEPIN

multiplicity: *single (static)*
type: *integer*

CHANGEPIN.

value	symbol
0	false
1	true

UGMASKM_WALK

multiplicity: *single (static)*
type: *integer*

WALK.

value	symbol
0	false
1	true

UGMASKM_ENGRESET

multiplicity: *single (static)*
type: *integer*

ENGRESET.

value	symbol
0	false
1	true

UGMASKM_DURESS

multiplicity: *single (static)*
type: *integer*

DURESS.

value	symbol
0	false
1	true

UGMASKM_TESTREP

multiplicity: *single (static)*
type: *integer*

TESTREP.

value	symbol
0	false
1	true

UGMASKM_COMM

multiplicity: *single (static)*
type: *integer*

COMM.

value	symbol
0	false
1	true

UGMASKM_CLEANER

multiplicity: *single (static)*
type: *integer*

CLEANER.

value	symbol
0	false
1	true

UGMASKM_AREA_LIST

multiplicity: *single (static)*
type: *integer*

AREA_LIST.

value	symbol
0	false
1	true

UGMASKM_MENUACC

multiplicity: *single (static)*
type: *integer*

MENUACC.

value	symbol
0	false
1	true

UGMASKM_INSTACC

multiplicity: *single (static)*
type: *integer*

INSTACC.

value	symbol
0	false
1	true

UGMASKM_VSTOP

multiplicity: *single (static)*
type: *integer*

VSTOP.

value	symbol
0	false
1	true

UGMASKM_SMSREP

multiplicity: *single (static)*
type: *integer*

SMSREP.

value	symbol
0	false
1	true

UGMASKM_SMSCTRL

multiplicity: *single (static)*
type: *integer*

SMSCTRL.

value	symbol
0	false
1	true

UGMASKM_NOPCLREP

multiplicity: *single (static)*
type: *integer*

NOPCLREP.

value	symbol
0	false
1	true

UGMASKM_LOGSACC

multiplicity: *single (static)*
type: *integer*

LOGSACC.

value	symbol
0	false
1	true

UGMASKM_DOORACCESS

multiplicity: *single (static)*
type: *integer*

DOORACCESS.

value	symbol
0	false
1	true

UserGroupMaskRas1

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 1.

UserGroupMaskRas2

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 2.

UserGroupMaskRas3

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 3.

UserGroupMaskRas4

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 4.

UserGroupMaskRas5

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 5.

UserGroupMaskRas6

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 6.

UserGroupMaskRas7

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 7.

UserGroupMaskRas8

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 8.

insertV.SACTActRASControl

direction: *output*

This is the outgoing call for "insertVSACTActRASControl" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

subindex

multiplicity: *single (static)*
type: *integer*

Schedule actions subindex.

value	symbol
1	void

RASCONTROLM_INDEX

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

RASCONTROLM_STATE

multiplicity: *single (static)*
type: *integer*

RAS state.

value	symbol
0	UNLOCK
1	LOCK

insertV.SACTActPSet1

direction: *output*

This is the outgoing call for "insertVSACTActPSet1" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

subindex

multiplicity: *single (static)*
type: *integer*

Schedule actions subindex.

value	symbol
1	void

PSET1M_AREAS.1

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 1.

PSET1M_AREAS.2

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 2.

PSET1M_AREAS.3

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 3.

PSET1M_AREAS.4

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 4.

PSET1M_AREAS.5

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 5.

PSET1M_AREAS.6

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 6.

PSET1M_AREAS.7

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 7.

PSET1M_AREAS.8

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 8.

PSET1M_AREAS

multiplicity: *single (static)*
type: *integer*

PartSet1 Areas.

insertV.SACTActPSet2

direction: *output*

This is the outgoing call for "insertVSACTActPSet2" method.

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

subindex

multiplicity: *single (static)*
type: *integer*

Schedule actions subindex.

value	symbol
1	void

PSET2M_AREAS.1

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 1.

PSET2M_AREAS.2

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 2.

PSET2M_AREAS.3

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 3.

PSET2M_AREAS.4

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 4.

PSET2M_AREAS.5

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 5.

PSET2M_AREAS.6

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 6.

PSET2M_AREAS.7

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 7.

PSET2M_AREAS.8

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 8.

PSET2M_AREAS

multiplicity: *single (static)*
type: *integer*

PartSet2 Areas.

insert.SchedActLst

direction: *output*

This is the outgoing call for "insertSchedActLst" method.

ACTIONLISTM_NAME

multiplicity: *single (static)*
type: *string*

Text.

ACTIONLISTM_ACTIVATION

multiplicity: *single (static)*
type: *integer*

Text.

value	symbol
0	false
1	true

ACTIONLISTM_ACTIONLIST1

multiplicity: *single (static)*
type: *integer*

Action list 1.

min	max
0	64

ACTIONLISTM_ACTIONLIST2

multiplicity: *single (static)*
type: *integer*

Action list 2.

min	max
0	64

ACTIONLISTM_ACTIONLIST3

multiplicity: *single (static)*
type: *integer*

Action list 3.

min	max
0	64

ACTIONLISTM_ACTIONLIST4

multiplicity: *single (static)*
type: *integer*

Action list 4.

min	max
0	64

ACTIONLISTM_ACTIONLIST5

multiplicity: *single (static)*
type: *integer*

Action list 5.

min	max
0	64

ACTIONLISTM_ACTIONLIST6

multiplicity: *single (static)*
type: *integer*

Action list 6.

min	max
0	64

ACTIONLISTM_ACTIONLIST7

multiplicity: *single (static)*
type: *integer*

Action list 7.

min	max
0	64

ACTIONLISTM_ACTIONLIST8

multiplicity: *single (static)*
type: *integer*

Action list 8.

min	max
0	64

index

multiplicity: *single (static)*
type: *integer*

Schedule action lists index.

min	max
1	32

insert.SchedExc

direction: *output*

This is the outgoing call for "insertSchedExc" method.

EXCEPTIONM_NAME

multiplicity: *single (static)*
type: *string*

Text.

EXCEPTIONM_SUBSTITUTE

multiplicity: *single (static)*
type: *integer*

Text.

value	symbol
0	false
1	true

EXCEPTIONM_ACTIVATION

multiplicity: *single (static)*
type: *integer*

Text.

value	symbol
0	false
1	true

EXCEPTIONM_REPEATTIME

multiplicity: *single (static)*
type: *integer*

Text.

value	symbol
0	FIXED_DATES

EXCEPTIONM_STARTMONTH

multiplicity: *single (static)*
type: *integer*

Text.

min	max
1	12

EXCEPTIONM_STARTDAY

multiplicity: *single (static)*
type: *integer*

Text.

min	max
1	31

EXCEPTIONM_STOPMONTH

multiplicity: *single (static)*
type: *integer*

Text.

min	max
1	12

EXCEPTIONM_STOPDAY

multiplicity: *single (static)*
type: *integer*

Text.

min	max
1	31

EXCEPTIONM_ACTION1

multiplicity: *single (static)*
type: *integer*

Exception action 1.

min	max
0	64

EXCEPTIONM_ACTION2

multiplicity: *single (static)*
type: *integer*

Exception action 2.

min	max
0	64

EXCEPTIONM_ACTION3

multiplicity: *single (static)*
type: *integer*

Exception action 3.

min	max
0	64

EXCEPTIONM_ACTION4

multiplicity: *single (static)*
type: *integer*

Exception action 4.

min	max
0	64

EXCEPTIONM_ACTIONLIST1

multiplicity: *single (static)*
type: *integer*

Exception action list 1.

min	max
0	32

EXCEPTIONM_ACTIONLIST2

multiplicity: *single (static)*
type: *integer*

Exception action list 2.

min	max
0	32

EXCEPTIONM_ACTIONLIST3

multiplicity: *single (static)*
type: *integer*

Exception action list 3.

min	max
0	32

EXCEPTIONM_ACTIONLIST4

multiplicity: *single (static)*
type: *integer*

Exception action list 4.

min	max
0	32

EXCEPTIONM_MONDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on mondays

value	symbol
0	false
1	true

EXCEPTIONM_TUESDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on tuesdays

value	symbol
0	false
1	true

EXCEPTIONM_WEDNESDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on wednesdays

value	symbol
0	false
1	true

EXCEPTIONM_THURSDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on thursdays

value	symbol
0	false
1	true

EXCEPTIONM_FRIDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on fridays

value	symbol
0	false
1	true

EXCEPTIONM_SATURDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on saturdays

value	symbol
0	false
1	true

EXCEPTIONM_SUNDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on sundays

value	symbol
0	false
1	true

EXCEPTIONM_YEAR

multiplicity: *single (static)*
type: *integer*

Exception valid on year

min	max
2000	2099

value	symbol
0	ANY

index

multiplicity: *single (static)*
type: *integer*

Schedule exceptions index.

min	max
1	64

insert.Schedule

direction: *output*

This is the outgoing call for "insertSchedule" method.

SCHEDULE_NAME

multiplicity: *single (static)*
type: *string*

Text.

SCHEDULEM_ACTION_MON1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Monday.

min	max
0	64

SCHEDULEM_ACTION_MON2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Monday.

min	max
0	64

SCHEDULEM_ACTION_MON3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Monday.

min	max
0	64

SCHEDULEM_ACTION_MON4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Monday.

min	max
0	64

SCHEDULEM_ACTION_TUE1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Tuesday.

min	max
0	64

SCHEDULEM_ACTION_TUE2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Tuesday.

min	max
0	64

SCHEDULEM_ACTION_TUE3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Tuesday.

min	max
0	64

SCHEDULEM_ACTION_TUE4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Tuesday.

min	max
0	64

SCHEDULEM_ACTION_WED1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Wednesday.

min	max
0	64

SCHEDULEM_ACTION_WED2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Wednesday.

min	max
0	64

SCHEDULEM_ACTION_WED3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Wednesday.

min	max
0	64

SCHEDULEM_ACTION_WED4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Wednesday.

min	max
0	64

SCHEDULEM_ACTION_THU1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Thursday.

min	max
0	64

SCHEDULEM_ACTION_THU2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Thursday.

min	max
0	64

SCHEDULEM_ACTION_THU3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Thursday.

min	max
0	64

SCHEDULEM_ACTION_THU4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Thursday.

min	max
0	64

SCHEDULEM_ACTION_FRI1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Friday.

min	max
0	64

SCHEDULEM_ACTION_FRI2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Friday.

min	max
0	64

SCHEDULEM_ACTION_FRI3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Friday.

min	max
0	64

SCHEDULEM_ACTION_FRI4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Friday.

min	max
0	64

SCHEDULEM_ACTION_SAT1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Saturday.

min	max
0	64

SCHEDULEM_ACTION_SAT2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Saturday.

min	max
0	64

SCHEDULEM_ACTION_SAT3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Saturday.

min	max
0	64

SCHEDULEM_ACTION_SAT4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Saturday.

min	max
0	64

SCHEDULEM_ACTION_SUN1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Sunday.

min	max
0	64

SCHEDULEM_ACTION_SUN2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Sunday.

min	max
0	64

SCHEDULEM_ACTION_SUN3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Sunday.

min	max
0	64

SCHEDULEM_ACTION_SUN4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Sunday.

min	max
0	64

SCHEDULEM_ACTIONLIST_MON1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Sunday.

min	max
0	32

SCHEDULEM_EXCEPTIONLIST1

multiplicity: *single (static)*
type: *integer*

Schedule exception list 1.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST2

multiplicity: *single (static)*
type: *integer*

Schedule exception list 2.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST3

multiplicity: *single (static)*
type: *integer*

Schedule exception list 3.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST4

multiplicity: *single (static)*
type: *integer*

Schedule exception list 4.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST5

multiplicity: *single (static)*
type: *integer*

Schedule exception list 5.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST6

multiplicity: *single (static)*
type: *integer*

Schedule exception list 6.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST7

multiplicity: *single (static)*
type: *integer*

Schedule exception list 7.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST8

multiplicity: *single (static)*
type: *integer*

Schedule exception list 8.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST9

multiplicity: *single (static)*
type: *integer*

Schedule exception list 9.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST10

multiplicity: *single (static)*
type: *integer*

Schedule exception list 10.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST11

multiplicity: *single (static)*
type: *integer*

Schedule exception list 11.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST12

multiplicity: *single (static)*
type: *integer*

Schedule exception list 12.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST13

multiplicity: *single (static)*
type: *integer*

Schedule exception list 13.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST14

multiplicity: *single (static)*
type: *integer*

Schedule exception list 14.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST15

multiplicity: *single (static)*
type: *integer*

Schedule exception list 15.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST16

multiplicity: *single (static)*
type: *integer*

Schedule exception list 16.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST17

multiplicity: *single (static)*
type: *integer*

Schedule exception list 17.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST18

multiplicity: *single (static)*
type: *integer*

Schedule exception list 18.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST19

multiplicity: *single (static)*
type: *integer*

Schedule exception list 19.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST20

multiplicity: *single (static)*
type: *integer*

Schedule exception list 20.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST21

multiplicity: *single (static)*
type: *integer*

Schedule exception list 21.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST22

multiplicity: *single (static)*
type: *integer*

Schedule exception list 22.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST23

multiplicity: *single (static)*
type: *integer*

Schedule exception list 23.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST24

multiplicity: *single (static)*
type: *integer*

Schedule exception list 24.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST25

multiplicity: *single (static)*
type: *integer*

Schedule exception list 25.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST26

multiplicity: *single (static)*
type: *integer*

Schedule exception list 26.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST27

multiplicity: *single (static)*
type: *integer*

Schedule exception list 27.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST28

multiplicity: *single (static)*
type: *integer*

Schedule exception list 28.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST29

multiplicity: *single (static)*
type: *integer*

Schedule exception list 29.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST30

multiplicity: *single (static)*
type: *integer*

Schedule exception list 30.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST31

multiplicity: *single (static)*
type: *integer*

Schedule exception list 31.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST32

multiplicity: *single (static)*
type: *integer*

Schedule exception list 32.

min	max
0	64

index

multiplicity: *single (static)*
type: *integer*

Schedules index.

min	max
1	4

insert.ScheduleDayActions

direction: *output*

This is the outgoing call for "insertScheduleDayActions" method.

day

multiplicity: *single (static)*
type: *integer*

Schedule day.

value	symbol
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday

SCHEDULEM_ACTION_1

multiplicity: *single (static)*
type: *integer*

Schedule action 1.

min	max
0	64

SCHEDULEM_ACTION_2

multiplicity: *single (static)*
type: *integer*

Schedule action 2.

min	max
0	64

SCHEDULEM_ACTION_3

multiplicity: *single (static)*
type: *integer*

Schedule action 3.

min	max
0	64

SCHEDULEM_ACTION_4

multiplicity: *single (static)*
type: *integer*

Schedule action 4.

min	max
0	64

SCHEDULEM_ACTION_5

multiplicity: *single (static)*
type: *integer*

Schedule action 5.

min	max
0	64

SCHEDULEM_ACTION_6

multiplicity: *single (static)*
type: *integer*

Schedule action 6.

min	max
0	64

SCHEDULEM_ACTION_7

multiplicity: *single (static)*
type: *integer*

Schedule action 7.

min	max
0	64

SCHEDULEM_ACTION_8

multiplicity: *single (static)*
type: *integer*

Schedule action 8.

min	max
0	64

SCHEDULEM_ACTION_9

multiplicity: *single (static)*
type: *integer*

Schedule action 9.

min	max
0	64

SCHEDULEM_ACTION_10

multiplicity: *single (static)*
type: *integer*

Schedule action 10.

min	max
0	64

SCHEDULEM_ACTION_11

multiplicity: *single (static)*
type: *integer*

Schedule action 11.

min	max
0	64

SCHEDULEM_ACTION_12

multiplicity: *single (static)*
type: *integer*

Schedule action 12.

min	max
0	64

SCHEDULEM_ACTION_13

multiplicity: *single (static)*
type: *integer*

Schedule action 13.

min	max
0	64

SCHEDULEM_ACTION_14

multiplicity: *single (static)*
type: *integer*

Schedule action 14.

min	max
0	64

SCHEDULEM_ACTION_15

multiplicity: *single (static)*
type: *integer*

Schedule action 15.

min	max
0	64

SCHEDULEM_ACTION_16

multiplicity: *single (static)*
type: *integer*

Schedule action 16.

min	max
0	64

index

multiplicity: *single (static)*
type: *integer*

Schedules index.

min	max
1	4

insert.Schedule2

direction: *output*

This is the outgoing call for "insertSchedule2" method.

index

multiplicity: *single (static)*
type: *integer*

Schedules index.

min	max
1	4

SCHEDULEM_NAME

multiplicity: *single (static)*
type: *string*

SCHEDULEM_ACTIONLIST_MON1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Sunday.

min	max
0	32

SCHEDULEM_EXCEPTIONLIST1

multiplicity: *single (static)*
type: *integer*

Schedule exception list 1.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST2

multiplicity: *single (static)*
type: *integer*

Schedule exception list 2.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST3

multiplicity: *single (static)*
type: *integer*

Schedule exception list 3.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST4

multiplicity: *single (static)*
type: *integer*

Schedule exception list 4.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST5

multiplicity: *single (static)*
type: *integer*

Schedule exception list 5.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST6

multiplicity: *single (static)*
type: *integer*

Schedule exception list 6.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST7

multiplicity: *single (static)*
type: *integer*

Schedule exception list 7.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST8

multiplicity: *single (static)*
type: *integer*

Schedule exception list 8.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST9

multiplicity: *single (static)*
type: *integer*

Schedule exception list 9.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST10

multiplicity: *single (static)*
type: *integer*

Schedule exception list 10.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST11

multiplicity: *single (static)*
type: *integer*

Schedule exception list 11.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST12

multiplicity: *single (static)*
type: *integer*

Schedule exception list 12.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST13

multiplicity: *single (static)*
type: *integer*

Schedule exception list 13.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST14

multiplicity: *single (static)*
type: *integer*

Schedule exception list 14.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST15

multiplicity: *single (static)*
type: *integer*

Schedule exception list 15.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST16

multiplicity: *single (static)*
type: *integer*

Schedule exception list 16.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST17

multiplicity: *single (static)*
type: *integer*

Schedule exception list 17.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST18

multiplicity: *single (static)*
type: *integer*

Schedule exception list 18.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST19

multiplicity: *single (static)*
type: *integer*

Schedule exception list 19.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST20

multiplicity: *single (static)*
type: *integer*

Schedule exception list 20.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST21

multiplicity: *single (static)*
type: *integer*

Schedule exception list 21.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST22

multiplicity: *single (static)*
type: *integer*

Schedule exception list 22.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST23

multiplicity: *single (static)*
type: *integer*

Schedule exception list 23.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST24

multiplicity: *single (static)*
type: *integer*

Schedule exception list 24.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST25

multiplicity: *single (static)*
type: *integer*

Schedule exception list 25.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST26

multiplicity: *single (static)*
type: *integer*

Schedule exception list 26.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST27

multiplicity: *single (static)*
type: *integer*

Schedule exception list 27.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST28

multiplicity: *single (static)*
type: *integer*

Schedule exception list 28.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST29

multiplicity: *single (static)*
type: *integer*

Schedule exception list 29.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST30

multiplicity: *single (static)*
type: *integer*

Schedule exception list 30.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST31

multiplicity: *single (static)*
type: *integer*

Schedule exception list 31.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST32

multiplicity: *single (static)*
type: *integer*

Schedule exception list 32.

min	max
0	64

SCHEDULEM_ACTIVATION

multiplicity: *single (static)*
type: *integer*

Schedule active switch

value	symbol
0	false
1	true

SCHEDULEM_STARTDATE

multiplicity: *single (static)*
type: *integer*

Start date for schedule

min	max
257	287
513	541
769	799
1025	1054
1281	1311
1537	1566
1793	1823
2049	2079

min	max
2305	2334
2561	2591
2817	2846
3073	3103

SCHEDULEM_STOPDATE

multiplicity: *single (static)*
type: *integer*

Stop date for schedule

min	max
257	287
513	541
769	799
1025	1054
1281	1311
1537	1566
1793	1823
2049	2079
2305	2334
2561	2591
2817	2846
3073	3103

SCHEDULEM_STARTMONTH

multiplicity: *single (static)*
type: *integer*

Start month for schedule

min	max
1	12

SCHEDULEM_STARTDAY

multiplicity: *single (static)*
type: *integer*

Start day for schedule

min	max
1	31

SCHEDULEM_STOPMONTH

multiplicity: *single (static)*
type: *integer*

Stop month for schedule

min	max
1	12

SCHEDULEM_STOPDAY

multiplicity: *single (static)*
type: *integer*

Stop day for schedule

min	max
1	31

return.SchedAct

direction: *input*

This is the return message for method "selectSchedAct".

index

multiplicity: *single (static)*
type: *integer*

Schedule actions index.

min	max
1	64

SACTM_NAME

multiplicity: *single (static)*
type: *string*

Text.

SACTM_STARTTIME

multiplicity: *single (static)*
type: *integer*

Text.

min	max
0	1439

SACTM_FILTER

multiplicity: *single (static)*
type: *integer*

Filter.

min	max
0	64

SACTM_ACTIVATION

multiplicity: *single (static)*
type: *integer*

Text.

value	symbol
0	false
1	true

return.ActNone

direction: *input*

This is the return message for method "selectV[object]ActNone".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

return.ActSet

direction: *input*

This is the return message for method "selectV[object]ActSet".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

SETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Set Area 1.

SETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Set Area 2.

SETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Set Area 3.

SETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Set Area 4.

SETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Set Area 5.

SETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Set Area 6.

SETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Set Area 7.

SETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Set Area 8.

SETM_AREAS

multiplicity: *single (static)*
type: *integer*

Set Areas.

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

return.ActUnset

direction: *input*

This is the return message for method "selectV[object]ActUnset".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

UNSETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Unset Area 1.

UNSETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Unset Area 2.

UNSETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Unset Area 3.

UNSETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Unset Area 4.

UNSETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Unset Area 5.

UNSETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Unset Area 6.

UNSETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Unset Area 7.

UNSETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Unset Area 8.

UNSETM_AREAS

multiplicity: *single (static)*
type: *integer*

Unset Areas.

return.ActTrigger

direction: *input*

This is the return message for method "selectV[object]ActTrigger".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

TRIGGERM_INDEX

multiplicity: *single (static)*
type: *integer*

Trigger index.

min	max
1	255

TRIGGERM_STATE

multiplicity: *single (static)*
type: *integer*

Trigger state.

value	symbol
0	CLEAR
1	SET
2	TOGGLE

return.ActDoorbell

direction: *input*

This is the return message for method "selectV[object]ActDoorbell".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

DOORBELLM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 1.

DOORBELLM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 2.

DOORBELLM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 3.

DOORBELLM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 4.

DOORBELLM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 5.

DOORBELLM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 6.

DOORBELLM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 7.

DOORBELLM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Doorbell Area 8.

DOORBELLM_AREAS

multiplicity: *single (static)*
type: *integer*

Doorbell Areas.

DOORBELLM_STATE

multiplicity: *single (static)*
type: *integer*

Doorbell state.

value	symbol
0	CLEAR
1	SET
2	TOGGLE

return.ActUGMask

direction: *input*

This is the return message for method "selectV[object]ActUGMask".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

UGMASKM_INDEX

multiplicity: *single (static)*
type: *integer*

User Group index.

min	max
1	16

UGMASKM_FULLSET

multiplicity: *single (static)*
type: *integer*

FULLSET.

value	symbol
0	false
1	true

UGMASKM_PARTSET

multiplicity: *single (static)*
type: *integer*

PARTSET.

value	symbol
0	false
1	true

UGMASKM_UNSET

multiplicity: *single (static)*
type: *integer*

UNSET.

value	symbol
0	false
1	true

UGMASKM_INHIBIT

multiplicity: *single (static)*
type: *integer*

INHIBIT.

value	symbol
0	false
1	true

UGMASKM_ISOLATE

multiplicity: *single (static)*
type: *integer*

ISOLATE.

value	symbol
0	false
1	true

UGMASKM_TIMEDATE

multiplicity: *single (static)*
type: *integer*

TIMEDATE.

value	symbol
0	false
1	true

UGMASKM_FSET

multiplicity: *single (static)*
type: *integer*

FSET.

value	symbol
0	false
1	true

UGMASKM_CHANGEPIN

multiplicity: *single (static)*
type: *integer*

CHANGEPIN.

value	symbol
0	false
1	true

UGMASKM_WALK

multiplicity: *single (static)*
type: *integer*

WALK.

value	symbol
0	false
1	true

UGMASKM_ENGRESET

multiplicity: *single (static)*
type: *integer*

ENGRESET.

value	symbol
0	false
1	true

UGMASKM_DURESS

multiplicity: *single (static)*
type: *integer*

DURESS.

value	symbol
0	false
1	true

UGMASKM_TESTREP

multiplicity: *single (static)*
type: *integer*

TESTREP.

value	symbol
0	false
1	true

UGMASKM_COMM

multiplicity: *single (static)*
type: *integer*

COMM.

value	symbol
0	false
1	true

UGMASKM_CLEANER

multiplicity: *single (static)*
type: *integer*

CLEANER.

value	symbol
0	false
1	true

UGMASKM_AREA_LIST

multiplicity: *single (static)*
type: *integer*

AREA_LIST.

value	symbol
0	false
1	true

UGMASKM_MENUACC

multiplicity: *single (static)*
type: *integer*

MENUACC.

value	symbol
0	false
1	true

UGMASKM_INSTACC

multiplicity: *single (static)*
type: *integer*

INSTACC.

value	symbol
0	false
1	true

UGMASKM_VSTOP

multiplicity: *single (static)*
type: *integer*

VSTOP.

value	symbol
0	false
1	true

UGMASKM_SMSREP

multiplicity: *single (static)*
type: *integer*

SMSREP.

value	symbol
0	false
1	true

UGMASKM_SMSCTRL

multiplicity: *single (static)*
type: *integer*

SMSCTRL.

value	symbol
0	false
1	true

UGMASKM_NOPCLREP

multiplicity: *single (static)*
type: *integer*

NOPCLREP.

value	symbol
0	false
1	true

UGMASKM_LOGSACC

multiplicity: *single (static)*
type: *integer*

LOGSACC.

value	symbol
0	false
1	true

UGMASKM_DOORACCESS

multiplicity: *single (static)*
type: *integer*

DOORACCESS.

value	symbol
0	false
1	true

UserGroupMaskRas1

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 1.

UserGroupMaskRas2

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 2.

UserGroupMaskRas3

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 3.

UserGroupMaskRas4

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 4.

UserGroupMaskRas5

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 5.

UserGroupMaskRas6

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 6.

UserGroupMaskRas7

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 7.

UserGroupMaskRas8

multiplicity: *single (static)*
type: *boolean*

UserGroupMaskRas 8.

return.ActTakePicture

direction: *input*

This is the return message for method "selectV[object]ActUGMask".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

CAMERA_ID

multiplicity: *single (static)*
type: *integer*

Camera ID.

min	max
1	128
257	368

CS_NUMBER

multiplicity: *single (static)*
type: *integer*

Central station number.

min	max
1	16

return.ActRASControl

direction: *input*

This is the return message for method "selectV[object]ActRASControl".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

RASCONTROLM_INDEX

multiplicity: *single (static)*
type: *integer*

RAS index.

min	max
1	8

RASCONTROLM_STATE

multiplicity: *single (static)*
type: *integer*

RAS state.

value	symbol
0	UNLOCK
1	LOCK

return.ActPSet1

direction: *input*

This is the return message for method "selectV[object]ActPSet1".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

PSET1M_AREAS.1

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 1.

PSET1M_AREAS.2

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 2.

PSET1M_AREAS.3

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 3.

PSET1M_AREAS.4

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 4.

PSET1M_AREAS.5

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 5.

PSET1M_AREAS.6

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 6.

PSET1M_AREAS.7

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 7.

PSET1M_AREAS.8

multiplicity: *single (static)*
type: *boolean*

PartSet1 Area 8.

PSET1M_AREAS

multiplicity: *single (static)*
type: *integer*

PartSet1 Areas.

return.ActPSet2

direction: *input*

This is the return message for method "selectV[object]ActPSet2".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

PSET2M_AREAS.1

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 1.

PSET2M_AREAS.2

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 2.

PSET2M_AREAS.3

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 3.

PSET2M_AREAS.4

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 4.

PSET2M_AREAS.5

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 5.

PSET2M_AREAS.6

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 6.

PSET2M_AREAS.7

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 7.

PSET2M_AREAS.8

multiplicity: *single (static)*
type: *boolean*

PartSet2 Area 8.

PSET2M_AREAS

multiplicity: *single (static)*
type: *integer*

PartSet2 Areas.

return.ActInh

direction: *input*

This is the return message for method "selectV[object]ActInh".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

return.ActTCall

direction: *input*

This is the return message for method "selectV[object]ActTCall".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

return.ActPCC

direction: *input*

This is the return message for method "selectV[object]ActPCC".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

return.ActServIn

direction: *input*

This is the return message for method "selectV[object]ActServIn".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

return.ActPanic

direction: *input*

This is the return message for method "selectV[object]ActPanic".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

return.ActDoorbellRAS

direction: *input*

This is the return message for method "selectV[object]ActDoorbellRAS".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

DOORBELLM_RAS.1

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 1.

DOORBELLM_RAS.2

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 2.

DOORBELLM_RAS.3

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 3.

DOORBELLM_RAS.4

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 4.

DOORBELLM_RAS.5

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 5.

DOORBELLM_RAS.6

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 6.

DOORBELLM_RAS.7

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 7.

DOORBELLM_RAS.8

multiplicity: *single (static)*
type: *boolean*

Doorbell RAS 8.

DOORBELLM_RAS

multiplicity: *single (static)*
type: *integer*

Doorbell Areas.

DOORBELLRASM_STATE

multiplicity: *single (static)*
type: *integer*

Doorbell state.

value	symbol
0	CLEAR
1	SET
2	TOGGLE

return.ActSetWET

direction: *input*

This is the return message for method "selectV[object]ActSetWET".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

SETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Set Area 1.

SETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Set Area 2.

SETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Set Area 3.

SETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Set Area 4.

SETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Set Area 5.

SETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Set Area 6.

SETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Set Area 7.

SETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Set Area 8.

SETM_AREAS

multiplicity: *single (static)*
type: *integer*

Set Areas.

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

return.ActFireReset

direction: *input*

This is the return message for method "selectV[object]ActFireReset".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

SETM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Set Area 1.

SETM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Set Area 2.

SETM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Set Area 3.

SETM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Set Area 4.

SETM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Set Area 5.

SETM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Set Area 6.

SETM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Set Area 7.

SETM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Set Area 8.

SETM_AREAS

multiplicity: *single (static)*
type: *integer*

Set Areas.

return.ActOpenZn

direction: *input*

This is the return message for method "selectV[object]ActOpenZn".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

return.ActAlarmZn

direction: *input*

This is the return message for method "selectV[object]ActAlrmZn".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

return.ActFaults

direction: *input*

This is the return message for method "selectV[object]ActFaults".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

return.ActAlarmMem

direction: *input*

This is the return message for method "selectV[object]ActAlarmMem".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

return.ActZonesAck

direction: *input*

This is the return message for method "selectV[object]ActZonesAck".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

return.ActWalkTest

direction: *input*

This is the return message for method "selectV[object]ActWalkTest".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

WALKTESTM_AREAS.1

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 1.

WALKTESTM_AREAS.2

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 2.

WALKTESTM_AREAS.3

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 3.

WALKTESTM_AREAS.4

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 4.

WALKTESTM_AREAS.5

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 5.

WALKTESTM_AREAS.6

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 6.

WALKTESTM_AREAS.7

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 7.

WALKTESTM_AREAS.8

multiplicity: *single (static)*
type: *boolean*

Walk Test Area 8.

WALKTESTM_AREAS

multiplicity: *single (static)*
type: *integer*

Walk Test Areas.

return.ActOutputTest

direction: *input*

This is the return message for method "selectV[object]ActOutputTest".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

UCODE

multiplicity: *single (static)*
type: *integer*

User code request

value	symbol
0	false
1	true

OUTPUTTESTM_OUTPUT.1

multiplicity: *single (static)*
type: *integer*

Output test 1.

min	max
1	200

value	symbol
0	NONE

OUTPUTTESTM_OUTPUT.2

multiplicity: *single (static)*
type: *integer*

Output test 2.

min	max
1	200

value	symbol
0	NONE

OUTPUTTESTM_OUTPUT.3

multiplicity: *single (static)*
type: *integer*

Output test 3.

min	max
1	200

value	symbol
0	NONE

OUTPUTTESTM_OUTPUT.4

multiplicity: *single (static)*
type: *integer*

Output test 4.

min	max
1	200

value	symbol
0	NONE

return.ActFire

direction: *input*

This is the return message for method "selectV[object]ActFire".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

return.ActMedical

direction: *input*

This is the return message for method "selectV[object]ActMedical".

index

multiplicity: *single (static)*
type: *integer*

Object index.

subindex

multiplicity: *single (static)*
type: *integer*

Object subindex.

return.SchedActLst

direction: *input*

This is the return message for method "selectSchedActLst".

ACTIONLISTM_NAME

multiplicity: *single (static)*
type: *string*

Text.

ACTIONLISTM_ACTIVATION

multiplicity: *single (static)*
type: *integer*

Text.

value	symbol
0	false
1	true

ACTIONLISTM_ACTIONLIST1

multiplicity: *single (static)*
type: *integer*

Action list 1.

min	max
0	64

ACTIONLISTM_ACTIONLIST2

multiplicity: *single (static)*
type: *integer*

Action list 2.

min	max
0	64

ACTIONLISTM_ACTIONLIST3

multiplicity: *single (static)*
type: *integer*

Action list 3.

min	max
0	64

ACTIONLISTM_ACTIONLIST4

multiplicity: *single (static)*
type: *integer*

Action list 4.

min	max
0	64

ACTIONLISTM_ACTIONLIST5

multiplicity: *single (static)*
type: *integer*

Action list 5.

min	max
0	64

ACTIONLISTM_ACTIONLIST6

multiplicity: *single (static)*
type: *integer*

Action list 6.

min	max
0	64

ACTIONLISTM_ACTIONLIST7

multiplicity: *single (static)*
type: *integer*

Action list 7.

min	max
0	64

ACTIONLISTM_ACTIONLIST8

multiplicity: *single (static)*
type: *integer*

Action list 8.

min	max
0	64

index

multiplicity: *single (static)*
type: *integer*

Schedule action lists index.

min	max
1	32

return.SchedExc

direction: *input*

This is the return message for method "selectSchedExc".

EXCEPTIONM_NAME

multiplicity: *single (static)*
type: *string*

Text.

EXCEPTIONM_SUBSTITUTE

multiplicity: *single (static)*
type: *integer*

Text.

value	symbol
0	false
1	true

EXCEPTIONM_ACTIVATION

multiplicity: *single (static)*
type: *integer*

Text.

value	symbol
0	false
1	true

EXCEPTIONM_REPEATTIME

multiplicity: *single (static)*
type: *integer*

Text.

value	symbol
0	FIXED_DATES

EXCEPTIONM_STARTMONTH

multiplicity: *single (static)*
type: *integer*

Text.

min	max
1	12

EXCEPTIONM_STARTDAY

multiplicity: *single (static)*
type: *integer*

Text.

min	max
1	31

EXCEPTIONM_STOPMONTH

multiplicity: *single (static)*
type: *integer*

Text.

min	max
1	12

EXCEPTIONM_STOPDAY

multiplicity: *single (static)*
type: *integer*

Text.

min	max
1	31

EXCEPTIONM_ACTION1

multiplicity: *single (static)*
type: *integer*

Exception action 1.

min	max
0	64

EXCEPTIONM_ACTION2

multiplicity: *single (static)*
type: *integer*

Exception action 2.

min	max
0	64

EXCEPTIONM_ACTION3

multiplicity: *single (static)*
type: *integer*

Exception action 3.

min	max
0	64

EXCEPTIONM_ACTION4

multiplicity: *single (static)*
type: *integer*

Exception action 4.

min	max
0	64

EXCEPTIONM_ACTIONLIST1

multiplicity: *single (static)*
type: *integer*

Exception action list 1.

min	max
0	32

EXCEPTIONM_ACTIONLIST2

multiplicity: *single (static)*
type: *integer*

Exception action list 2.

min	max
0	32

EXCEPTIONM_ACTIONLIST3

multiplicity: *single (static)*
type: *integer*

Exception action list 3.

min	max
0	32

EXCEPTIONM_ACTIONLIST4

multiplicity: *single (static)*
type: *integer*

Exception action list 4.

min	max
0	32

EXCEPTIONM_MONDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on mondays

value	symbol
0	false
1	true

EXCEPTIONM_TUESDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on tuesdays

value	symbol
0	false
1	true

EXCEPTIONM_WEDNESDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on wednesdays

value	symbol
0	false
1	true

EXCEPTIONM_THURSDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on thursdays

value	symbol
0	false
1	true

EXCEPTIONM_FRIDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on fridays

value	symbol
0	false
1	true

EXCEPTIONM_SATURDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on saturdays

value	symbol
0	false
1	true

EXCEPTIONM_SUNDAY

multiplicity: *single (static)*
type: *integer*

Exception valid on sundays

value	symbol
0	false
1	true

EXCEPTIONM_YEAR

multiplicity: *single (static)*
type: *integer*

Exception valid on year

min	max
2000	2099

value	symbol
0	ANY

index

multiplicity: *single (static)*
type: *integer*

Schedule exceptions index.

min	max
1	64

return.Schedule

direction: *input*

This is the return message for method "selectSchedule".

SCHEDULE_NAME

multiplicity: *single (static)*
type: *string*

Text.

SCHEDULEM_ACTION_MON1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Monday.

min	max
0	64

SCHEDULEM_ACTION_MON2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Monday.

min	max
0	64

SCHEDULEM_ACTION_MON3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Monday.

min	max
0	64

SCHEDULEM_ACTION_MON4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Monday.

min	max
0	64

SCHEDULEM_ACTION_TUE1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Tuesday.

min	max
0	64

SCHEDULEM_ACTION_TUE2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Tuesday.

min	max
0	64

SCHEDULEM_ACTION_TUE3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Tuesday.

min	max
0	64

SCHEDULEM_ACTION_TUE4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Tuesday.

min	max
0	64

SCHEDULEM_ACTION_WED1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Wednesday.

min	max
0	64

SCHEDULEM_ACTION_WED2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Wednesday.

min	max
0	64

SCHEDULEM_ACTION_WED3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Wednesday.

min	max
0	64

SCHEDULEM_ACTION_WED4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Wednesday.

min	max
0	64

SCHEDULEM_ACTION_THU1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Thursday.

min	max
0	64

SCHEDULEM_ACTION_THU2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Thursday.

min	max
0	64

SCHEDULEM_ACTION_THU3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Thursday.

min	max
0	64

SCHEDULEM_ACTION_THU4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Thursday.

min	max
0	64

SCHEDULEM_ACTION_FRI1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Friday.

min	max
0	64

SCHEDULEM_ACTION_FRI2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Friday.

min	max
0	64

SCHEDULEM_ACTION_FRI3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Friday.

min	max
0	64

SCHEDULEM_ACTION_FRI4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Friday.

min	max
0	64

SCHEDULEM_ACTION_SAT1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Saturday.

min	max
0	64

SCHEDULEM_ACTION_SAT2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Saturday.

min	max
0	64

SCHEDULEM_ACTION_SAT3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Saturday.

min	max
0	64

SCHEDULEM_ACTION_SAT4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Saturday.

min	max
0	64

SCHEDULEM_ACTION_SUN1

multiplicity: *single (static)*
type: *integer*

Schedule action 1 for Sunday.

min	max
0	64

SCHEDULEM_ACTION_SUN2

multiplicity: *single (static)*
type: *integer*

Schedule action 2 for Sunday.

min	max
0	64

SCHEDULEM_ACTION_SUN3

multiplicity: *single (static)*
type: *integer*

Schedule action 3 for Sunday.

min	max
0	64

SCHEDULEM_ACTION_SUN4

multiplicity: *single (static)*
type: *integer*

Schedule action 4 for Sunday.

min	max
0	64

SCHEDULEM_ACTIONLIST_MON1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Sunday.

min	max
0	32

SCHEDULEM_EXCEPTIONLIST1

multiplicity: *single (static)*
type: *integer*

Schedule exception list 1.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST2

multiplicity: *single (static)*
type: *integer*

Schedule exception list 2.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST3

multiplicity: *single (static)*
type: *integer*

Schedule exception list 3.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST4

multiplicity: *single (static)*
type: *integer*

Schedule exception list 4.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST5

multiplicity: *single (static)*
type: *integer*

Schedule exception list 5.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST6

multiplicity: *single (static)*
type: *integer*

Schedule exception list 6.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST7

multiplicity: *single (static)*
type: *integer*

Schedule exception list 7.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST8

multiplicity: *single (static)*
type: *integer*

Schedule exception list 8.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST9

multiplicity: *single (static)*
type: *integer*

Schedule exception list 9.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST10

multiplicity: *single (static)*
type: *integer*

Schedule exception list 10.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST11

multiplicity: *single (static)*
type: *integer*

Schedule exception list 11.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST12

multiplicity: *single (static)*
type: *integer*

Schedule exception list 12.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST13

multiplicity: *single (static)*
type: *integer*

Schedule exception list 13.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST14

multiplicity: *single (static)*
type: *integer*

Schedule exception list 14.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST15

multiplicity: *single (static)*
type: *integer*

Schedule exception list 15.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST16

multiplicity: *single (static)*
type: *integer*

Schedule exception list 16.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST17

multiplicity: *single (static)*
type: *integer*

Schedule exception list 17.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST18

multiplicity: *single (static)*
type: *integer*

Schedule exception list 18.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST19

multiplicity: *single (static)*
type: *integer*

Schedule exception list 19.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST20

multiplicity: *single (static)*
type: *integer*

Schedule exception list 20.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST21

multiplicity: *single (static)*
type: *integer*

Schedule exception list 21.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST22

multiplicity: *single (static)*
type: *integer*

Schedule exception list 22.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST23

multiplicity: *single (static)*
type: *integer*

Schedule exception list 23.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST24

multiplicity: *single (static)*
type: *integer*

Schedule exception list 24.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST25

multiplicity: *single (static)*
type: *integer*

Schedule exception list 25.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST26

multiplicity: *single (static)*
type: *integer*

Schedule exception list 26.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST27

multiplicity: *single (static)*
type: *integer*

Schedule exception list 27.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST28

multiplicity: *single (static)*
type: *integer*

Schedule exception list 28.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST29

multiplicity: *single (static)*
type: *integer*

Schedule exception list 29.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST30

multiplicity: *single (static)*
type: *integer*

Schedule exception list 30.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST31

multiplicity: *single (static)*
type: *integer*

Schedule exception list 31.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST32

multiplicity: *single (static)*
type: *integer*

Schedule exception list 32.

min	max
0	64

index

multiplicity: *single (static)*
type: *integer*

Schedules index.

min	max
1	4

return.ScheduleDayActions

direction: *input*

This is the outgoing call for "selectScheduleDayActions" method.

day

multiplicity: *single (static)*
type: *integer*

Schedule day.

value	symbol
1	Monday
2	Tuesday
3	Wednesday
4	Thursday
5	Friday
6	Saturday
7	Sunday

SCHEDULEM_ACTION_1

multiplicity: *single (static)*
type: *integer*

Schedule action 1.

min	max
0	64

SCHEDULEM_ACTION_2

multiplicity: *single (static)*
type: *integer*

Schedule action 2.

min	max
0	64

SCHEDULEM_ACTION_3

multiplicity: *single (static)*
type: *integer*

Schedule action 3.

min	max
0	64

SCHEDULEM_ACTION_4

multiplicity: *single (static)*
type: *integer*

Schedule action 4.

min	max
0	64

SCHEDULEM_ACTION_5

multiplicity: *single (static)*
type: *integer*

Schedule action 5.

min	max
0	64

SCHEDULEM_ACTION_6

multiplicity: *single (static)*
type: *integer*

Schedule action 6.

min	max
0	64

SCHEDULEM_ACTION_7

multiplicity: *single (static)*
type: *integer*

Schedule action 7.

min	max
0	64

SCHEDULEM_ACTION_8

multiplicity: *single (static)*
type: *integer*

Schedule action 8.

min	max
0	64

SCHEDULEM_ACTION_9

multiplicity: *single (static)*
type: *integer*

Schedule action 9.

min	max
0	64

SCHEDULEM_ACTION_10

multiplicity: *single (static)*
type: *integer*

Schedule action 10.

min	max
0	64

SCHEDULEM_ACTION_11

multiplicity: *single (static)*
type: *integer*

Schedule action 11.

min	max
0	64

SCHEDULEM_ACTION_12

multiplicity: *single (static)*
type: *integer*

Schedule action 12.

min	max
0	64

SCHEDULEM_ACTION_13

multiplicity: *single (static)*
type: *integer*

Schedule action 13.

min	max
0	64

SCHEDULEM_ACTION_14

multiplicity: *single (static)*
type: *integer*

Schedule action 14.

min	max
0	64

SCHEDULEM_ACTION_15

multiplicity: *single (static)*
type: *integer*

Schedule action 15.

min	max
0	64

SCHEDULEM_ACTION_16

multiplicity: *single (static)*
type: *integer*

Schedule action 16.

min	max
0	64

index

multiplicity: *single (static)*
type: *integer*

Schedules index.

min	max
1	4

return.Schedule2

direction: *input*

This is the outgoing call for "selectSchedule2" method.

index

multiplicity: *single (static)*
type: *integer*

Schedules index.

min	max
1	4

SCHEDULEM_NAME

multiplicity: *single (static)*
type: *string*

SCHEDULEM_ACTIONLIST_MON1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_MON4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Monday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_TUE4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Tuesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_WED4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Wednesday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_THU4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Thursday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_FRI4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Friday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SAT4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Saturday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN1

multiplicity: *single (static)*
type: *integer*

Schedule action list 1 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN2

multiplicity: *single (static)*
type: *integer*

Schedule action list 2 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN3

multiplicity: *single (static)*
type: *integer*

Schedule action list 3 for Sunday.

min	max
0	32

SCHEDULEM_ACTIONLIST_SUN4

multiplicity: *single (static)*
type: *integer*

Schedule action list 4 for Sunday.

min	max
0	32

SCHEDULEM_EXCEPTIONLIST1

multiplicity: *single (static)*
type: *integer*

Schedule exception list 1.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST2

multiplicity: *single (static)*
type: *integer*

Schedule exception list 2.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST3

multiplicity: *single (static)*
type: *integer*

Schedule exception list 3.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST4

multiplicity: *single (static)*
type: *integer*

Schedule exception list 4.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST5

multiplicity: *single (static)*
type: *integer*

Schedule exception list 5.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST6

multiplicity: *single (static)*
type: *integer*

Schedule exception list 6.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST7

multiplicity: *single (static)*
type: *integer*

Schedule exception list 7.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST8

multiplicity: *single (static)*
type: *integer*

Schedule exception list 8.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST9

multiplicity: *single (static)*
type: *integer*

Schedule exception list 9.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST10

multiplicity: *single (static)*
type: *integer*

Schedule exception list 10.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST11

multiplicity: *single (static)*
type: *integer*

Schedule exception list 11.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST12

multiplicity: *single (static)*
type: *integer*

Schedule exception list 12.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST13

multiplicity: *single (static)*
type: *integer*

Schedule exception list 13.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST14

multiplicity: *single (static)*
type: *integer*

Schedule exception list 14.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST15

multiplicity: *single (static)*
type: *integer*

Schedule exception list 15.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST16

multiplicity: *single (static)*
type: *integer*

Schedule exception list 16.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST17

multiplicity: *single (static)*
type: *integer*

Schedule exception list 17.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST18

multiplicity: *single (static)*
type: *integer*

Schedule exception list 18.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST19

multiplicity: *single (static)*
type: *integer*

Schedule exception list 19.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST20

multiplicity: *single (static)*
type: *integer*

Schedule exception list 20.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST21

multiplicity: *single (static)*
type: *integer*

Schedule exception list 21.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST22

multiplicity: *single (static)*
type: *integer*

Schedule exception list 22.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST23

multiplicity: *single (static)*
type: *integer*

Schedule exception list 23.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST24

multiplicity: *single (static)*
type: *integer*

Schedule exception list 24.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST25

multiplicity: *single (static)*
type: *integer*

Schedule exception list 25.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST26

multiplicity: *single (static)*
type: *integer*

Schedule exception list 26.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST27

multiplicity: *single (static)*
type: *integer*

Schedule exception list 27.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST28

multiplicity: *single (static)*
type: *integer*

Schedule exception list 28.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST29

multiplicity: *single (static)*
type: *integer*

Schedule exception list 29.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST30

multiplicity: *single (static)*
type: *integer*

Schedule exception list 30.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST31

multiplicity: *single (static)*
type: *integer*

Schedule exception list 31.

min	max
0	64

SCHEDULEM_EXCEPTIONLIST32

multiplicity: *single (static)*
type: *integer*

Schedule exception list 32.

min	max
0	64

SCHEDULEM_ACTIVATION

multiplicity: *single (static)*
type: *integer*

Schedule active switch

value	symbol
0	false
1	true

SCHEDULEM_STARTDATE

multiplicity: *single (static)*
type: *integer*

Start date for schedule

min	max
257	287
513	541
769	799
1025	1054
1281	1311
1537	1566
1793	1823
2049	2079

min	max
2305	2334
2561	2591
2817	2846
3073	3103

SCHEDULEM_STOPDATE

multiplicity: *single (static)*
type: *integer*

Stop date for schedule

min	max
257	287
513	541
769	799
1025	1054
1281	1311
1537	1566
1793	1823
2049	2079
2305	2334
2561	2591
2817	2846
3073	3103

SCHEDULEM_STARTMONTH

multiplicity: *single (static)*
type: *integer*

Start month for schedule

min	max
1	12

SCHEDULEM_STARTDAY

multiplicity: *single (static)*
type: *integer*

Start day for schedule

min	max
1	31

SCHEDULEM_STOPMONTH

multiplicity: *single (static)*
type: *integer*

Stop month for schedule

min	max
1	12

SCHEDULEM_STOPDAY

multiplicity: *single (static)*
type: *integer*

Stop day for schedule

min	max
1	31

prepareEnc.UPLOAD

direction: *output*

This message is used to prepare the encapsulated transmission of database from panel to PC.

The response is return.short (operation status)

startEnc.UPLOAD

direction: *output*

This message is used to start the encapsulated transmission of database from panel to PC.

The response is return.void

sessionID

multiplicity: *single (static)*
type: *integer*

Session

event.Aggregate

direction: *input*

This message is used to return the encapsulated transmission of database from panel to PC.

typeID

multiplicity: *single (static)*
type: *integer*

TypeID.

sessionID

multiplicity: *single (static)*
type: *integer*

Session ID

prepareEnc.DOWNLOAD

direction: *output*

call.Aggregate

direction: *input*

This message is used to transmit encapsulated database configuration from PC to panel.

return.Aggregate

direction: *input*

This message is used to transmit encapsulated database configuration from PC to panel.

finishedEnc.UPLOAD

direction: *input*

sessionID

multiplicity: *single (static)*
type: *integer*

status

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

value	symbol
1	FAULT_METHOD_ERROR
2	FAULT_NO_ACCESS
3	FAULT_NO_OBJECT
4	FAULT_LOG_NEWEST
5	FAULT_LOG_OLDEST
6	FAULT_PIN
7	FAULT_CARD
8	FAULT_USER
9	FAULT_UG
10	FAULT_UG_DATA
11	FAULT_UG_AREAS
12	FAULT_UG_PRIVILEGES
13	FAULT_COMMIT
14	FAULT_USER_UG_NOT_EXIST
15	FAULT_INSTALLER_UG
16	FAULT_SUPERVISOR_UG
17	FAULT_USER_UG
18	FAULT_ZONE_DATA
19	FAULT_UG_EVENT_FILTER
20	FAULT_DATA_NOT_VALID
21	FAULT_OFFLINE
22	FAULT_SUPERVISOR_REQUIRED
23	FAULT_PANEL_BUSY
24	FAULT_CC_WRONG_STAT
25	FAULT_CC_WRONG_TYPE
26	FAULT_CC_WRONG_SESSION
27	FAULT_CC_WRONG_AREAS
28	FAULT_CC_BUSY_AREAS
29	FAULT_CC_WRONG_PRIVILEGES

value	symbol
30	FAULT_CC_OPERATION_CANCELED
31	FAULT_CC_COMMUNICATION_TOUT
32	FAULT_CC_BUSY_PEB
33	FAULT_CC_NRDY_PIC
34	FAULT_CC_WRONG_TSTAMP
35	FAULT_CC_OFFL_PEB
36	FAULT_CC_COMM_PEB
37	FAULT_CC_BUSY_CAMERA
38	FAULT_CC_ISOL_CAMERA
39	FAULT_CC_LIMIT_PIC
40	FAULT_CC_NOTALLOWED_PIC
41	FAULT_CC_FAULT
42	FAULT_CC_MEMORYFULL
50	FAULT_FEATURE_NOT_SUPPORTED
51	FAULT_RETRY
65535	METHOD_NOT_FOUND

cancelEncUPLOAD

direction: *output*

sessionID

multiplicity: *single (static)*
type: *integer*

Session

event.PebMemoryStatus

direction: *input*

This is the return message for method "select.Zone".

typeID

multiplicity: *single (static)*
type: *integer*

TypeID.

sessionID

multiplicity: *single (static)*
type: *integer*

Session ID

pebIndex

multiplicity: *single (static)*
type: *integer*

Index of the PEB.

pictures

multiplicity: *single (static)*
type: *integer*

Number of pictures on the PEB.

total_mem

multiplicity: *single (static)*
type: *integer*

Total memory on the PEB.

free_mem

multiplicity: *single (static)*
type: *integer*

Free memory on the PEB.

event.PebMemoryClearProgress

direction: *input*

This is a progress notification of clearing PEB memory and associated WPCs.

This message is used during PEB/WPC clear memory session.

typeID

multiplicity: *single (static)*
type: *integer*

TypeID.

sessionID

multiplicity: *single (static)*
type: *integer*

Session ID

pebIndex

multiplicity: *single (static)*
type: *integer*

PEB index which is currently under clear memory procedure.

progress

multiplicity: *single (static)*
type: *integer*

Progress of clear memory procedure in %.

min	max
0	100

return.GetPebs

direction: *input*

This is the return message for method "msgCamera.GetPebs".

pebName.1

multiplicity: *single (static)*
type: *string*

Camera Name

pebName.2

multiplicity: *single (static)*
type: *string*

Camera Name

pebName.3

multiplicity: *single (static)*
type: *string*

Camera Name

pebName.4

multiplicity: *single (static)*
type: *string*

Camera Name

pebIndex.1

multiplicity: *single (static)*
type: *integer*

Camera index

pebIndex.2

multiplicity: *single (static)*
type: *integer*

Camera index

pebIndex.3

multiplicity: *single (static)*
type: *integer*

Camera index

pebIndex.4

multiplicity: *single (static)*
type: *integer*

Camera index

return.GetPebCameras

direction: *input*

This is the return message for method "msgCamera.GetPebCameras".

cameraName.1

multiplicity: *single (static)*
type: *string*

Camera Name

cameraName.2

multiplicity: *single (static)*
type: *string*

Camera Name

cameraName.3

multiplicity: *single (static)*
type: *string*

Camera Name

cameraName.4

multiplicity: *single (static)*
type: *string*

Camera Name

cameraName.5

multiplicity: *single (static)*
type: *string*

Camera Name

cameraName.6

multiplicity: *single (static)*
type: *string*

Camera Name

cameraName.7

multiplicity: *single (static)*
type: *string*

Camera Name

cameraName.8

multiplicity: *single (static)*
type: *string*

Camera Name

cameraIndex.1

multiplicity: *single (static)*
type: *integer*

Camera index

cameraIndex.2

multiplicity: *single (static)*
type: *integer*

Camera index

cameraIndex.3

multiplicity: *single (static)*
type: *integer*

Camera index

cameraIndex.4

multiplicity: *single (static)*
type: *integer*

Camera index

cameraIndex.5

multiplicity: *single (static)*
type: *integer*

Camera index

cameraIndex.6

multiplicity: *single (static)*
type: *integer*

Camera index

cameraIndex.7

multiplicity: *single (static)*
type: *integer*

Camera index

cameraIndex.8

multiplicity: *single (static)*
type: *integer*

Camera index

msgCamera.GetPEBs

direction: *output*

This message is used to return the encapsulated transmission of database from panel to PC.

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

msgCamera.GetPebCameras

direction: *output*

This message is used to return the encapsulated transmission of database from panel to PC.

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

pebIndex

multiplicity: *single (static)*
type: *integer*

pebIndex

event.CameraDir

direction: *input*

This message is used to return the encapsulated transmission of database from panel to PC.

typeID

multiplicity: *single (static)*
type: *integer*

TypeID.

sessionID

multiplicity: *single (static)*
type: *integer*

Session ID

pictureID

multiplicity: *single (static)*
type: *integer*

pictureID

size

multiplicity: *single (static)*
type: *integer*

size of the picture

time

multiplicity: *single (static)*
type: *datetime*

The timestamp value when the picture was taken.

Remarks

- It is assumed that the value uses Coordinated Universal Time format aka *UTC*.
- Leap seconds available in *UTC* are not supported.

format: `date+time`

zone

multiplicity: *single (static)*
type: *integer*

zone

eventType

multiplicity: *single (static)*
type: *integer*

eventType

value	symbol
1	BA
2	TA
3	F
4	FA
5	PA
6	MA
7	D1
8	D2
9	WALKTEST
10	TEST_CALL
11	DOWNLOADER
12	SMS
13	NOT_REQUESTED

imageSequence

multiplicity: *single (static)*
type: *integer*

imageSequence

event.PictureChunk

direction: *input*

This message is used to return the encapsulated transmission of database from panel to PC.

typeID

multiplicity: *single (static)*
type: *integer*

TypeID.

sessionID

multiplicity: *single (static)*
type: *integer*

Session ID

offset

multiplicity: *single (static)*
type: *integer*

Size of the picture.

bufferLength

multiplicity: *single (static)*
type: *integer*

Length of buffer.

pictureChunkData

multiplicity: *single (static)*
type: *string*

RAW Picture chunk data.

sesCamera.prepareCameraDir

direction: *output*

This message is used to prepare the encapsulated transmission of database from panel to PC.

The response is return.short (operation status)

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

cameraIndex

multiplicity: *single (static)*
type: *integer*

cameraIndex

sesCamera.prepareLoadPicture

direction: *output*

This message is used to prepare the encapsulated transmission of database from panel to PC.

The response is return.short (operation status)

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

cameraIndex

multiplicity: *single (static)*
type: *integer*

cameraIndex

pictureIndex

multiplicity: *single (static)*
type: *integer*

pictureIndex

time

multiplicity: *single (static)*
type: *datetime*

The timestamp value when the picture was taken.

Remarks

- It is assumed that the value uses Coordinated Universal Time format aka *UTC*.
- Leap seconds available in *UTC* are not supported.

format: `date+time`

sesCamera.prepareTakePicture

direction: *output*

This message is used to prepare the encapsulated transmission of database from panel to PC.

The response is return.short (operation status)

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

cameraIndex

multiplicity: *single (static)*
type: *integer*

cameraIndex

sesCamera.prepareMemoryClear

direction: *output*

This message is used to prepare the encapsulated transmission of database from panel to PC.

The response is return.short (operation status)

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

pebIndex

multiplicity: *single (static)*
type: *integer*

pebIndex

sesCamera.prepareMemoryInfo

direction: *output*

This message is used to return the encapsulated transmission of database from panel to PC.

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

pebIndex

multiplicity: *single (static)*
type: *integer*

pebIndex

sesCamera.start

direction: *output*

This message is used to start the encapsulated transmission of database from panel to PC.

The response is return.void

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

pictSessionID

multiplicity: *single (static)*
type: *integer*

Picture session ID

sesCamera.completed

direction: *input*

sessionID

multiplicity: *single (static)*
type: *integer*

status

multiplicity: *single (static)*
type: *integer*
nullable: *yes*

value	symbol
1	FAULT_METHOD_ERROR
2	FAULT_NO_ACCESS
3	FAULT_NO_OBJECT
4	FAULT_LOG_NEWEST
5	FAULT_LOG_OLDEST
6	FAULT_PIN
7	FAULT_CARD
8	FAULT_USER
9	FAULT_UG
10	FAULT_UG_DATA
11	FAULT_UG_AREAS
12	FAULT_UG_PRIVILEGES
13	FAULT_COMMIT
14	FAULT_USER_UG_NOT_EXIST
15	FAULT_INSTALLER_UG
16	FAULT_SUPERVISOR_UG
17	FAULT_USER_UG
18	FAULT_ZONE_DATA
19	FAULT_UG_EVENT_FILTER
20	FAULT_DATA_NOT_VALID
21	FAULT_OFFLINE
22	FAULT_SUPERVISOR_REQUIRED
23	FAULT_PANEL_BUSY
24	FAULT_CC_WRONG_STAT
25	FAULT_CC_WRONG_TYPE
26	FAULT_CC_WRONG_SESSION
27	FAULT_CC_WRONG_AREAS
28	FAULT_CC_BUSY_AREAS
29	FAULT_CC_WRONG_PRIVILEGES

value	symbol
30	FAULT_CC_OPERATION_CANCELED
31	FAULT_CC_COMMUNICATION_TOUT
32	FAULT_CC_BUSY_PEB
33	FAULT_CC_NRDY_PIC
34	FAULT_CC_WRONG_TSTAMP
35	FAULT_CC_OFFL_PEB
36	FAULT_CC_COMM_PEB
37	FAULT_CC_BUSY_CAMERA
38	FAULT_CC_ISOL_CAMERA
39	FAULT_CC_LIMIT_PIC
40	FAULT_CC_NOTALLOWED_PIC
41	FAULT_CC_FAULT
42	FAULT_CC_MEMORYFULL
50	FAULT_FEATURE_NOT_SUPPORTED
51	FAULT_RETRY
65535	METHOD_NOT_FOUND

sesCamera.cancel

direction: *output*

sessionID

multiplicity: *single (static)*
type: *integer*

Session identifier (16 bit).

pictSessionID

multiplicity: *single (static)*
type: *integer*

Session

Index of messages

add.RemoteUsers 445
add.UserGroups 628
add.Users 441
begin.changeSessionKey 38, 16, 39, 40
begin.InitKey 1077
blockID.Area 864
blockID.Camera 882
blockID.CEvFilter 870
blockID.CS 872
blockID.DGP 866
blockID.DL 873
blockID.Fob 881
blockID.Master 884
blockID.Output 868
blockID.PCC 875
blockID.RAS 865
blockID.RemoteUser 883
blockID.SchedAct 877
blockID.SchedActLst 878
blockID.SchedExc 879
blockID.Schedule 880
blockID.SiaEvent 876
blockID.SYS 874
blockID.Trigger 869
blockID.User 867
blockID.UserGroup 871
blockID.Zone 863
blockIDM.Area 886
blockIDM.Camera 904
blockIDM.CEvFilter 892
blockIDM.CS 894
blockIDM.DGP 888
blockIDM.DL 895
blockIDM.Fob 903
blockIDM.Output 890
blockIDM.PCC 897
blockIDM.RAS 887
blockIDM.RemoteUser 905
blockIDM.SchedAct 899
blockIDM.SchedActLst 900
blockIDM.SchedExc 901
blockIDM.Schedule 902
blockIDM.SiaEvent 898
blockIDM.SYS 896
blockIDM.Trigger 891
blockIDM.User 889
blockIDM.UserGroup 893
blockIDM.Zone 885
call.Aggregate 1252
cancelEncUPLOAD 1256
close.LOG 46, 16, 45
commit.Users 1032
createCC.A_CONFAL 212, 18
createCC.A_PARTSET 222, 18
createCC.A_PARTSET2 227, 18
createCC.A_SET 207, 18
createCC.A_STATE 204, 19
createCC.A_UNSET 218, 19
createCC.A_WALKTST 215, 18
createCC.CAM_RANGETST 261, 19
createCC.DEVICE 235, 19
createCC.ENG_RES 244, 19
createCC.OUT_TRIG 238, 19
createCC.OUTPUT 241, 19
createCC.PC_CONN 250, 19
createCC.PICTURES 259
createCC.SYS_INV_WALKTST 264, 19
createCC.SYS_WALKTST_MODE 267
createCC.TEST_CALL 256, 19
createCC.TIME_DATE 247, 19
createCC.USER 253, 19
createCC.ZONE 232, 19
delete.Camera 580
delete.CEvFilter 822
delete.DGP 818
delete.FOB 817
delete.Output 820
delete.RAS 816
delete.User 819
delete.UserGroup 821
delete.Zone 815
deleteM.Camera 832
deleteM.CEvFilter 847
deleteM.DGP 835
deleteM.FOB 829
deleteM.Output 841
deleteM.RAS 826
deleteM.User 838
deleteM.UserGroup 844
deleteM.Zone 823
destroyCC.SESSION 270, 19, 204, 204, 207, 208, 208, 209, 210, 210, 212, 212, 212, 212, 215, 215, 215, 216, 218, 218, 218, 218, 219, 219, 222, 223, 223, 224, 225, 225, 227, 228, 228, 229, 230, 230, 232, 232, 235, 235, 238, 238, 241, 241, 244, 244, 244, 247, 247, 250, 250, 253, 253, 256, 256, 256, 256, 259, 261, 261, 261, 261, 264, 264, 264, 264, 267, 267
device.Description 32, 16, 20, 31
device.disconnect 43, 16
device.getConnect 41, 16
device.getDescription 31, 15, 16, 20, 32
device.panelId 28, 29
device.SecondPIN 1018
end.changeSessionKey 39, 16, 38
event.Aggregate 1250
event.CameraDir 1266
event.PebMemoryClearProgress 1258
event.PebMemoryStatus 1257
event.PictureChunk 1268
fault 22
finishedEnc.UPLOAD 1254
fnCC.A_CONFAL_CONFALARM 288, 212
fnCC.A_CONFAL_GETALARM 287, 212
fnCC.A_CONFAL_START 286, 212, 212
fnCC.A_SET_FORCEDSET 285, 208, 208, 209, 209, 223, 223, 224, 224, 228, 228, 229, 229
fnCC.A_SET_GETACTIVE 281, 209, 209, 224, 224, 229, 229
fnCC.A_SET_GETFAULT 280, 208, 208, 223, 223, 228, 228
fnCC.A_SET_GETINHIB 282, 209, 209, 224, 224, 229, 229
fnCC.A_SET_INHACTIVE 284, 209, 209, 224, 224, 229, 229
fnCC.A_SET_INHFAULT 283, 208, 208, 223, 223, 228, 228
fnCC.A_SET_SETAREAS 279, 208, 208, 209, 210, 223, 223, 224, 225, 228, 228, 229, 230

fnCC.A_STATE_GET_INH 277, 204
fnCC.A_STATE_GET_UNINH 278, 204
fnCC.A_UNSET_CONFALARM 302, 218
fnCC.A_UNSET_CONFFAULT 304, 219
fnCC.A_UNSET_GETALARM 301, 218
fnCC.A_UNSET_GETFAULT 303, 219
fnCC.A_UNSET_SKIP 300, 218, 219
fnCC.A_UNSET_UNSETAREAS 299, 218, 218
fnCC.A_WALKTST_ADD_ZONE 298, 215, 215
fnCC.A_WALKTST_GET_WARN_TIME 293
fnCC.A_WALKTST_GETEV 291, 215
fnCC.A_WALKTST_GETLIST 290, 215, 215
fnCC.A_WALKTST_GETRES 292, 216, 216
fnCC.A_WALKTST_START 289, 215
fnCC.A_WALKTST_START_WITH_REP 297, 215
fnCC.BATTERY_TEST_CANCEL 315, 235
fnCC.BATTERY_TEST_START 313, 235
fnCC.CAM_RANGETST_ADDCAM 276, 261
fnCC.CAM_RANGETST_START 275, 261
fnCC.DEVICE_ISOLATE 309, 235
fnCC.DEVICE_UNISOLATE 311, 235
fnCC.ENG_RES_DORESET 321, 244
fnCC.ENG_RES_GETCODE 323, 244
fnCC.ENG_RES_GETRESULT 322, 244, 244
fnCC.OUT_SCHED_TRIG_ACTIVATE 333
fnCC.OUT_SCHED_TRIG_DEACTIVATE 334
fnCC.OUT_TRIG_ACTIVATE 317, 238
fnCC.OUT_TRIG_DEACTIVATE 318, 238
fnCC.OUTPUT_ACTIVATE 319, 241
fnCC.OUTPUT_DEACTIVATE 320, 241
fnCC.PC_CONN_START 325, 250
fnCC.PC_CONN_STOP 326, 250
fnCC.SYS_CHANGE_WALKTST_MODE 296, 267
fnCC.SYS_INV_WALKTST_REP 294, 264
fnCC.SYS_INV_WALKTST_RESET 295, 264
fnCC.TEST_CALL_START 335, 256, 256
fnCC.TEST_CALL_STATUS 336, 256, 256, 256, 256
fnCC.TIME_DATE_SET 324, 247
fnCC.USER_GETPHONE 329, 253
fnCC.USER_SETCONTROL 327, 253
fnCC.USER_SETPHONE 331, 253
fnCC.USER_SETPIN 332, 253
fnCC.USER_SETREPORT 328, 253
fnCC.ZONE_INHIBIT 307, 232
fnCC.ZONE_ISOLATE 305, 232
fnCC.ZONE_UNINHIBIT 308, 232
fnCC.ZONE_UNISOLATE 306, 232
generate.userPIN 1028
generate.userRemotePIN 1029
get.liveEvents 99, 17
get.privileges 89, 17
get.timedate 87, 17
get.UserInfo 95, 17
getAvailM.Camera 862
getAvailM.CEvFilter 859
getAvailM.DGP 856
getAvailM.FOB 861
getAvailM.Output 858
getAvailM.RAS 855
getAvailM.User 857
getAvailM.UserGroup 860
getAvailM.Zone 854
getCOS.AREA 112, 77
getCOS.CAMERA 140, 81
getCOS.CS 126, 79
getCOS.DGP 116, 78
getCOS.EXCP 135, 80
getCOS.FILTER 120, 79
getCOS.FOB 138, 80
getCOS.OUT 118, 78
getCOS.PCC 123, 79
getCOS.RAS 114, 77
getCOS.TRIGG 129, 80
getCOS.UG 133, 79
getCOS.USER 131, 78
getCOS.ZONE 109, 77
getSTAT.AREA 147, 77, 113
getSTAT.CAMERA 202, 81, 141
getSTAT.CS 184, 79, 127
getSTAT.DGP 165, 78, 117
getSTAT.DGP0 169, 78
getSTAT.EXCP 194, 80, 136
getSTAT.FILTER 176, 79, 121
getSTAT.FOB 199, 80, 139
getSTAT.OUT 174, 78, 119
getSTAT.PCC 178, 79, 124
getSTAT.RAS 161, 77, 115
getSTAT.SCAL 196, 80
getSTAT.SYS 180, 79
getSTAT.TRIGG 186, 80, 130
getSTAT.UG 192, 79, 134
getSTAT.USER 189, 78, 132
getSTAT.ZONE 142, 77, 110
getValid.Areas 850
getValid.Cameras 851
insert.Area 427
insert.Camera 581
insert.CEvFilter 766
insert.CS 469
insert.CS_2 474
insert.CS_CMN 480
insert.CSAccount 1037
insert.CSAccount2 1038
insert.DGP 592
insert.DL 675
insert.DL_2 1045
insert.DL_75XX 1070
insert.DL_CHIRON 1068
insert.DL_GSM 1056
insert.DL_IP 1061
insert.DL_ISDN 1053
insert.DL_PSTN 1050
insert.DL_STEL 1066
insert.FOB 558
insert.InitKey 1078
insert.Output 599
insert.panelId 30
insert.PCC 706
insert.PCC_2 708
insert.PCC_CMN 711
insert.putCARD 1021
insert.putPIN 1019
insert.putRemotePIN 1020
insert.RAS 503
insert.SchedAct 1087
insert.SchedActLst 1109
insert.SchedExc 1112
insert.Schedule 1118
insert.Schedule2 1141
insert.ScheduleDayActions 1137
insert.SiaEvent 681
insert.SYS1 947
insert.SYS2 962
insert.SYS3 989
insert.SYS4 1015
insert.Trigger 604
insert.User 437
insert.UserGroup 617

insert.Zone 376
insertV.CS_IP 488
insertV.CS_PHONE 487
insertV.CS_USER 489
insertV.CS_USERGROUP 490
insertV.DL_GSM_IP 1072
insertV.DL_GSM_MMS 1075
insertV.FOBActNone 560
insertV.FOBActPanic 570
insertV.FOBActPSet1 566
insertV.FOBActPSet2 568
insertV.FOBActSet 561
insertV.FOBActTakePicture 571
insertV.FOBActTrigger 565
insertV.FOBActUnset 563
insertV.PCC_IP 714
insertV.PCC_PHONE 713
insertV.RASActAlarmMem 545
insertV.RASActAlarmZn 543
insertV.RASActDoorbell 520
insertV.RASActDoorbellRAS 534
insertV.RASActFaults 544
insertV.RASActFire 551
insertV.RASActFireReset 540
insertV.RASActInh 529
insertV.RASActMedical 552
insertV.RASActNone 513
insertV.RASActOpenZn 542
insertV.RASActOutputTest 549
insertV.RASActPanic 533
insertV.RASActPCC 531
insertV.RASActPSet1 523
insertV.RASActPSet2 526
insertV.RASActServIn 532
insertV.RASActSet 514
insertV.RASActSetWET 537
insertV.RASActTakePicture 553
insertV.RASActTCall 530
insertV.RASActTrigger 519
insertV.RASActUnset 517
insertV.RASActWalkTest 547
insertV.RASActZonesAck 546
insertV.SACTActDoorbell 1094
insertV.SACTActPSet1 1105
insertV.SACTActPSet2 1107
insertV.SACTActRASControl 1104
insertV.SACTActSet 1089
insertV.SACTActTrigger 1093
insertV.SACTActUGMask 1097
insertV.SACTActUnset 1091
is.Alive 44
msg.MONITOR 64, 17, 48
msgCamera.GetPebCameras 1265
msgCamera.GetPEBs 1264
msgCOS.ALL 77, 17
msgCOS.CAM_RANGETST 82, 17
msgCOS.SYS_INV_WALKTST_REP 86, 17
open.LOG 45, 16
pause.MONITOR 63, 17
prepareEnc.DOWNLOAD 1251
prepareEnc.UPLOAD 1248
return.ActAlarmMem 1193
return.ActAlarmZn 1191
return.ActDoorbell 1164
return.ActDoorbellRAS 1184
return.ActFaults 1192
return.ActFire 1199
return.ActFireReset 1188
return.ActInh 1179
return.ActMedical 1200
return.ActNone 1158
return.ActOpenZn 1190
return.ActOutputTest 1197
return.ActPanic 1183
return.ActPCC 1181
return.ActPSet1 1175
return.ActPSet2 1177
return.ActRASControl 1174
return.ActServIn 1182
return.ActSet 1159
return.ActSetWET 1186
return.ActTakePicture 1173
return.ActTCall 1180
return.ActTrigger 1163
return.ActUGMask 1166
return.ActUnset 1161
return.ActWalkTest 1195
return.ActZonesAck 1194
return.Aggregate 1253
return.Area 423
return.AreaNames 340
return.AvailMCamera 914
return.AvailMCevFilter 911
return.AvailMDGP 908
return.AvailMFob 913
return.AvailMOutput 910
return.AvailMRAS 907
return.AvailMUser 909
return.AvailMUserGroup 912
return.AvailMZone 906
return.BlockId 915, 30, 1078
return.BlockIdAreaM 920
return.BlockIdCameraM 938
return.BlockIdCevFilterM 926
return.BlockIdCSM 928
return.BlockIdDGPM 922
return.BlockIdDLM 929
return.BlockIdFobM 937
return.BlockIdMaster 916
return.BlockIdOutputM 924
return.BlockIdPCCM 931
return.BlockIdRASM 921
return.BlockIdRemoteUserM 939
return.BlockIdSchedActLstM 934
return.BlockIdSchedActM 933
return.BlockIdSchedExcM 935
return.BlockIdScheduleM 936
return.BlockIdSiaEventM 932
return.BlockIdSysM 930
return.BlockIdTriggerM 925
return.BlockIdUserGroupM 927
return.BlockIdUserM 923
return.BlockIdZoneM 919
return.bool 26, 61, 62, 63, 275, 276, 279, 283, 284, 285, 286, 288, 289, 294, 295, 296, 297, 298, 299, 302, 304, 305, 306, 307, 308, 309, 311, 313, 315, 317, 318, 319, 320, 322, 324, 325, 326, 327, 328, 331, 332, 335
return.Camera 573
return.CameraNames 374
return.CEvFilter 717
return.CEvFilterNames 350
return.changeSessionKey 40, 38
return.CommandStatus 715
return.CS 448
return.CS_2 453
return.CS_CMN 459
return.CS_IP 466
return.CS_PHONE 465

return.CS_USER 467
return.CS_USERGROUP 468
return.CSAccount 1034
return.CSAccount2 1035
return.CSNames 354
return.DGP 589
return.DGPNames 344
return.DL 644
return.DL_2 1040
return.DL_75XX 667
return.DL_CHIRON 665
return.DL_GSM 653
return.DL_GSM_IP 670
return.DL_GSM_MMS 673
return.DL_INFO 643, 642
return.DL_IP 658
return.DL_ISDN 650
return.DL_PSTN 647
return.DL_STEL 663
return.DL_VEMPTY 669
return.DLNames 356
return.FOB 556
return.FobNames 372
return.getCARD 1026
return.getLOG 48, 47, 64
return.GetPebCameras 1261
return.GetPebs 1259
return.getPIN 1025
return.getRemotePIN 1027
return.Output 596
return.OutputNames 348
return.panelId 29, 28
return.PCC 697
return.PCC_2 699
return.PCC_CMN 702
return.PCC_IP 705
return.PCC_PHONE 704
return.PCCNames 360
return.privileges 90, 89
return.RAS 493
return.RASNames 342
return.SchedAct 1156
return.SchedActLst 1201
return.SchedActLstNames 366
return.SchedActNames 364
return.SchedExc 1204
return.SchedExcNames 368
return.Schedule 1210
return.Schedule2 1233
return.ScheduleDayActions 1229
return.ScheduleNames 370
return.short 25, 204, 207, 212, 215, 218, 222, 227, 232, 235, 238, 241, 244, 247, 250, 253, 256, 259, 261, 264, 267, 293, 323, 336
return.SiaEvent 688
return.statusCC 272, 271, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 321, 322, 323, 335, 336
return.SYS0 1008
return.SYS1 941
return.SYS2 954
return.SYS3 971
return.SYS4 1012
return.sysevent 103, 99, 208, 209, 223, 224, 228, 229, 277, 278, 280, 281, 282, 287, 290, 291, 292, 301, 303
return.SYSNames 358
return.timedate 88, 87
return.Trigger 603
return.TriggerNames 362
return.User 433
return.UserGroup 606
return.UserGroupNames 352
return.UserInfo 96, 95
return.UserNames 346
return.UserPhone 330, 329
return.validAreas 852, 850
return.validCameras 853, 851
return.void 24, 38, 39, 44, 270, 277, 278, 280, 281, 282, 287, 290, 291, 292, 300, 301, 303, 321
return.Zone 414
return.ZoneNames 338
returnCOS.AREA 113, 112
returnCOS.CAMERA 141, 140
returnCOS.CS 127, 126
returnCOS.DGP 117, 116
returnCOS.EXCP 136, 135
returnCOS.FILTER 121, 120
returnCOS.FOB 139, 138
returnCOS.OUT 119, 118
returnCOS.PCC 124, 123
returnCOS.RAS 115, 114
returnCOS.TRIGG 130, 129
returnCOS.UG 134, 133
returnCOS.USER 132, 131
returnCOS.ZONE 110, 109
returnSTAT.AREA 148, 147
returnSTAT.CAMERA 203, 202
returnSTAT.CS 185, 184
returnSTAT.DGP 166, 165
returnSTAT.DGP0 170, 169
returnSTAT.EXCP 195, 194
returnSTAT.FILTER 177, 176
returnSTAT.FOB 200, 199
returnSTAT.OUT 175, 174
returnSTAT.PCC 179, 178
returnSTAT.RAS 162, 161
returnSTAT.SCAL 197, 196
returnSTAT.SYS 181, 180
returnSTAT.TRIGG 187, 186
returnSTAT.UG 193
returnSTAT.USER 190, 189
returnSTAT.ZONE 143, 142
select.Area 431
select.AreaNames 339
select.Camera 572
select.CameraNames 373
select.CEvFilter 716
select.CEvFilterNames 349
select.CS 446
select.CSAccount 1033
select.CSNames 353
select.DGP 588
select.DGPNames 343
select.DL 639
select.DL_INFO 642, 643
select.DLNames 355
select.FOB 554
select.FobNames 371
select.getCARD 1024
select.getLOG 47, 16, 45
select.getPIN 1022
select.getRemotePIN 1023
select.Output 595
select.OutputNames 347
select.PCC 695
select.PCCNames 359
select.RAS 491

select.RASNames 341
select.SchedAct 1080
select.SchedActLst 1082
select.SchedActLstNames 365
select.SchedActNames 363
select.SchedExc 1083
select.SchedExcNames 367
select.Schedule 1084
select.Schedule2 1086
select.ScheduleDayActions 1085
select.ScheduleNames 369
select.SiaEvent 678
select.SYS0 1007
select.SYS1 940
select.SYS2 953
select.SYS3 970
select.SYS4 1011
select.SYSNames 357
select.Trigger 602
select.TriggerNames 361
select.User 432
select.UserGroup 605
select.UserGroupNames 351
select.UserNames 345
select.Zone 375
select.ZoneNames 337
selectV.CS 447
selectV.DL 640
selectV.DL_MMS 641
selectV.FobAct 555
selectV.PCC 696
selectV.RASAct 492
selectV.SACTAct 1081
sesCamera.cancel 1277
sesCamera.completed 1275
sesCamera.prepareCameraDir 1269
sesCamera.prepareLoadPicture 1270
sesCamera.prepareMemoryClear 1272
sesCamera.prepareMemoryInfo 1273
sesCamera.prepareTakePicture 1271
sesCamera.start 1274
start.MONITOR 61, 17
start.Users 1030
startEnc.UPLOAD 1249
statusCC.SESSION 271, 19, 204, 204, 207, 208, 208, 209, 210, 210, 212, 212, 212, 212, 215, 215, 215, 216, 218, 218, 218, 218, 219, 219, 222, 223, 223, 224, 225, 225, 227, 228, 228, 229, 230, 230, 232, 232, 235, 235, 238, 238, 241, 241, 244, 244, 244, 247, 247, 250, 250, 253, 253, 256, 256, 256, 256, 261, 261, 261, 261, 264, 264, 264, 264, 267, 267
stop.MONITOR 62, 17
stop.Users 1031