A MITEL
PRODUCT
GUIDE

Unify OpenScape 4000

Import/Export (XIE) API

Service Documentation
08/2024

P31003-H34B0-S102-02-7620

Mitel

Notices

The information contained in this document is believed to be accurate in all respects but is not warranted by Mitel Europe Limited. The information is subject to change without notice and should not be construed in any way as a commitment by Mitel or any of its affiliates or subsidiaries. Mitel and its affiliates and subsidiaries assume no responsibility for any errors or omissions in this document. Revisions of this document or new editions of it may be issued to incorporate such changes. No part of this document can be reproduced or transmitted in any form or by any means - electronic or mechanical - for any purpose without written permission from Mitel Networks Corporation.

Trademarks

The trademarks, service marks, logos, and graphics (collectively “Trademarks”) appearing on Mitel’s Internet sites or in its publications are registered and unregistered trademarks of Mitel Networks Corporation (MNC) or its subsidiaries (collectively “Mitel), Unify Software and Solutions GmbH & Co. KG or its affiliates (collectively “Unify”) or others. Use of the Trademarks is prohibited without the express consent from Mitel and/or Unify. Please contact our legal department at iplegal@mitel.com for additional information. For a list of the worldwide Mitel and Unify registered trademarks, please refer to the website: http://www.mitel.com/trademarks.

© Copyright 2024, Mitel Networks Corporation

All rights reserved

Contents

1 General Information....12
1.1 Scope of Function....12
1.2 About This Manual....13
1.2.1 System Requirements....14
1.2.2 Hardware Requirements....14
1.2.3 Software Requirements....15

2 Installation....16
2.1 Server Installation....16
2.1.1 Server restrictions....16
2.2 Client Installation....16
2.2.1 Installing the Programmers Interface (DLL)....16
2.2.2 Installing the Export Table Client and api2hipath Tool....17

3 Server-Side Components....18
3.1 Directory structure....18
3.2 File Interface....18
3.2.1 dmsied....19
3.2.2 File Interface Configuration....19
3.2.3 File Interface Protocol File....20
3.3 Server side for Export Table client, api2hipath and user programs based on API Class Programmers interface....20
3.3.1 xieserver....20
3.3.1.1 xieserver Configuration....21
3.3.1.2 XIE-API Server Log File....21
3.4 Server Side for WebService Interface....22
3.5 Trace/Debug options for server-side components....22
3.6 Object Description Files (ODFs)....23
3.6.1 Customer Specific Fields....23
3.7 Delta Update Flag in CDB "admin" Table....24

4 Assigning Import/Export API Access Rights via OpenScape 4000 Manager/Assistant Account Management....25

5 XIE-API Class Library....28
5.1 OpenScape 4000 Manager/Assistant XIE-API Class Library Overview....28
5.2 Access to the XIE-API Class Library....28
5.3 XIE-API Class Library Call....28
5.4 Project Settings....29
5.5 Run-time Requirements....29
5.6 XIE-API Class Library Architecture....30
5.7 XIE-API Class Library OpenScape 4000 Manager Objects....30
5.8 XIE-API Class Library Export Functionality....30
5.9 XIE-API Class Library Import Functionality....30
5.10 XIE-API Security....31
5.11 XIE-API Class Library Object Files....31
5.12 XIE-API Class Library Key Values....31
5.13 XIE-API Class Library Comparison Operation....32
5.14 Classes....32
5.15 Class XIECommunication....33
5.15.1 Class XIECommunication Public Methods....33
5.15.2 Class XIECommunication Example....36

5.16 Class XIEdbInstance..... 37
5.16.1 Class XIEdbInstance Public Methods..... 37
5.17 Class XIEQueryInstance..... 39
5.17.1 Class XIEQueryInstance Public Methods..... 39
5.17.2 Class XIEQueryInstance Example..... 45
5.18 Class XIECreateInstance..... 47
5.18.1 Class XIECreateInstance Public Methods..... 47
5.18.2 Class XIECreateInstance Example..... 50
5.19 Class XIEDeleteInstance..... 51
5.19.1 Class XIEDeleteInstance Public Methods..... 51
5.19.2 Class XIEDeleteInstance Example..... 54
5.20 Class XIEModifyInstance..... 55
5.20.1 Class XIEModifyInstance Public Methods..... 56
5.20.2 Class XIEModifyInstance Example..... 60
5.21 Class XIEDeltaQueryInstance..... 61
5.21.1 Class XIEDeltaQueryInstance Public Methods..... 62
5.21.2 Class XIEDeltaQueryInstance Example..... 66

6 File Interface..... 69
6.1 Basic Concepts..... 69
6.2 Interface Description..... 70
6.2.1 Object Description Files (ODF)..... 70
6.2.2 Request and Response Files..... 70
6.3 Data Exchange Examples..... 73
6.3.1 Personal Identification Number (PIN)..... 74
6.3.2 Personal Data..... 75
6.3.3 Changing Display Names..... 76
6.3.4 Verifying Integration at OpenScape 4000 System..... 77
6.3.5 Derived ODFs..... 79

7 API Import/Export Client..... 82
7.1 General Information..... 82
7.2 Export Table Page..... 83
7.2.1 The File Menu..... 83
7.2.2 The Edit Menu..... 84
7.2.3 The View Menu..... 84
7.2.4 The Help Menu..... 84
7.2.5 The Tool Bar..... 85
7.2.6 Export Table Property Page Window..... 85
7.3 Import Table Page..... 88
7.3.1 The File Menu..... 89
7.3.2 The Edit Menu..... 89
7.3.3 The View Menu..... 89
7.3.4 The Help Menu..... 89
7.3.5 The Tool Bar..... 89
7.3.6 Import Table Property Page Window..... 90
7.4 Scheduler Page..... 93
7.4.1 The File Menu..... 93
7.4.2 The Edit Menu..... 93
7.4.3 The View Menu..... 93
7.4.4 The Help Menu..... 93
7.4.5 The Tool Bar..... 93
7.4.6 Scheduler Property Page Window..... 94

8 The api2hipath Command Line Tool..... 97
8.1 Exporting Data to an ASCII File..... 97
8.2 Delta-Export Data to an ASCII File..... 99

8.3 Importing Data from an ASCII File....100
8.4 Return values of api2hipath....102

9 XIE Web Service....103
9.1 Access to the interface....103
9.2 Security....104
9.3 List of web service operations....104
9.3.1 Object Data Files info....105
9.3.2 Object Data Files request control....105
9.3.3 Simple requests (without ODF)....108
9.3.4 Special requests used by CMP/UC....108
9.3.5 Simple Types....110
9.3.5.1 ActionType....110
9.3.5.2 ErrCode....111
9.3.5.3 RecordType....111
9.3.5.4 Status....111
9.3.5.5 SyncType....111
9.3.6 Complex Types....112
9.3.6.1 ColumnRecord....112
9.3.6.2 GetFullyQualifiedNumberRequest....112
9.3.6.3 GetFullyQualifiedNumberResult....112
9.3.6.4 RecordStatus....113
9.3.6.5 RequestDataRecord....113
9.3.6.6 Request....114
9.3.6.7 ResultCodeStruct....115
9.3.6.8 ResultData....115
9.3.6.9 ResultDataRecord....115
9.3.6.10 ResultStatus....116
9.3.6.11 XIEFailure....116
9.4 Client demo example....117
9.5 How-to write a Client....117
9.5.1 Requirements....117
9.5.2 How to open a secure session on the OpenScape system....118
9.5.3 Sequence of operations in user scenario....118
9.5.4 Example of usage of processData() operation....120
9.5.4.1 Select Statement....120
9.5.4.2 Insert statement....121
9.5.4.3 Example of usage of getStatus() operation....122
9.5.4.4 Example of usage of fetch/fetchNext operation....123

10 Description of ODFs....125
10.1 List of ODFs....125
10.1.1 OpenScape 4000 Objects....125
10.1.2 File Interface Objects....130
10.1.3 Assistant Objects....132
10.1.4 OpenScape/HiPath Inventory Management Objects....133
10.1.5 OpenScape/HiPath Accounting Management Objects....135
10.2 Detailed Description of ODFs....136
10.2.1 OpenScape 4000 Objects - Detailed Description....136
10.2.1.1 ACTION_CONTROL....136
10.2.1.2 ALARMS....138
10.2.1.3 APPEARANCE....139
10.2.1.4 APPEARANCE_ON....139
10.2.1.5 ATM_BOARD....140
10.2.1.6 BERUM....140
10.2.1.7 BUEND....141

10.2.1.8 BUILDING..... 142
10.2.1.9 BUTTON_INDIV..... 142
10.2.1.10 BUTTON_UPD..... 146
10.2.1.11 BUTTON_STANDARD..... 147
10.2.1.12 CFW..... 154
10.2.1.13 CHDMAIN..... 155
10.2.1.14 CHESE..... 157
10.2.1.15 CHESECOMB..... 158
10.2.1.16 CHESE_SUB..... 158
10.2.1.17 CHESE_TEAM..... 159
10.2.1.18 CMI_BASE_STAT..... 162
10.2.1.19 CMI_BOARD..... 163
10.2.1.20 CMI_BOARD_SUB..... 166
10.2.1.21 CMI_QUER..... 166
10.2.1.22 CMI_SYSTEM..... 166
10.2.1.23 CMI_SYSTEM_SUB..... 167
10.2.1.24 CODED_VAL..... 168
10.2.1.25 COEX..... 168
10.2.1.26 COMPANY_CODE..... 168
10.2.1.27 COMPORG..... 169
10.2.1.28 COS..... 169
10.2.1.29 COSXT..... 181
10.2.1.30 CTRYNUM..... 182
10.2.1.31 DEVCOMB..... 183
10.2.1.32 DEVCONST..... 184
10.2.1.33 DIMSU..... 185
10.2.1.34 DISPLAY_NAME..... 186
10.2.1.35 DOMAIN..... 186
10.2.1.36 DOWNLOAD_CLASS..... 187
10.2.1.37 DOWNLOAD_PARAMS..... 187
10.2.1.38 DOWNLOAD_SUPP..... 188
10.2.1.39 DYNAMIC_KEYDAT..... 190
10.2.1.40 E164_NUM..... 191
10.2.1.41 ERRORS..... 191
10.2.1.42 FIRMDAT..... 192
10.2.1.43 HUNTGRP..... 193
10.2.1.44 HUNTGRP_AC..... 193
10.2.1.45 HUNTGRP_DPLN_CPS..... 194
10.2.1.46 HUNTGRP_SERVICE..... 195
10.2.1.47 HUNTGRP_SUB..... 196
10.2.1.48 KNPRE..... 197
10.2.1.49 KZNR..... 197
10.2.1.50 KZNR_LIST..... 198
10.2.1.51 LBER..... 198
10.2.1.52 LCOS_DATA..... 198
10.2.1.53 LCOS_VOICE..... 201
10.2.1.54 LCR..... 204
10.2.1.55 LCRAUTH..... 209
10.2.1.56 LCRDPLN..... 210
10.2.1.57 LCRDPLN_SUB..... 210
10.2.1.58 LCRRULE..... 213
10.2.1.59 LCRRULE_SUB..... 213
10.2.1.60 LCRSCHED..... 214
10.2.1.61 LCRZONE..... 219
10.2.1.62 LOCDAT..... 220
10.2.1.63 LRTG..... 220

10.2.1.64 LRTG_SUB ... 221
10.2.1.65 NUMBERING_PLAN ... 222
10.2.1.66 ORDERS ... 223
10.2.1.67 ORG ... 224
10.2.1.68 PEN ... 224
10.2.1.69 PERSDAT ... 226
10.2.1.70 PERSDAT_VNR ... 229
10.2.1.71 PORTPERS/PERSPORT ... 233
10.2.1.72 PICKUPGRP ... 243
10.2.1.73 PICKUP_LINK ... 245
10.2.1.74 PICKUP_SUB ... 245
10.2.1.75 PIN ... 246
10.2.1.76 PINCHECK ... 248
10.2.1.77 PINCLASS ... 248
10.2.1.78 PINGENERAL ... 250
10.2.1.79 PINPERS (pin/persdat Join) ... 251
10.2.1.80 PORT ... 252
10.2.1.81 PRIV_DIAL_NUMBER ... 262
10.2.1.82 PRIV_DIAL_NUM_RA ... 263
10.2.1.83 PRIV_DIAL_NUM_ST ... 264
10.2.1.84 PRIV_DIAL_NUM_WV ... 264
10.2.1.85 ROUTE_INFO_WN ... 265
10.2.1.86 ROUTE_INFO_WT ... 265
10.2.1.87 SBS_PORT ... 266
10.2.1.88 SHELF_DATA ... 274
10.2.1.89 SPEC_SUBSCRIBER ... 275
10.2.1.90 STATION_RANGE ... 277
10.2.1.91 STATION_XREF ... 278
10.2.1.92 SWITCH ... 278
10.2.1.93 TRANS_SERVICE ... 283
10.2.1.94 UC_ACTION_LOG ... 283
10.2.1.95 UPD_PERSDAT ... 285
10.2.1.96 UPD_PIN ... 285
10.2.1.97 UPD_PORT ... 285
10.2.1.98 VBZ ... 286
10.2.1.99 VBZ_AUSWEIS ... 294
10.2.1.100 VBZGR ... 302
10.2.1.101 VMSCOS ... 303
10.2.1.102 VOICE_MAIL ... 303
10.2.1.103 XPR_DATA ... 304
10.2.2 File Interface Objects - Detailed Description ... 307
10.2.2.1 BUILDIMP ... 307
10.2.2.2 COMPIMP ... 308
10.2.2.3 COSDATA ... 308
10.2.2.4 CTRYCODES ... 309
10.2.2.5 DEVCOM ... 309
10.2.2.6 DOMAINDATA ... 310
10.2.2.7 EXTENSIONS ... 310
10.2.2.8 GRP_PBOOK_DATA ... 311
10.2.2.9 HUNT ... 312
10.2.2.10 LOCIMP ... 312
10.2.2.11 ORGIMP ... 313
10.2.2.12 PENDATA ... 313
10.2.2.13 PERS ... 314
10.2.2.14 PICKUP ... 316
10.2.2.15 PINDATA ... 316

10.2.2.16 PORTDATA 317
10.2.2.17 PORT_DISP (port/display_name Join) 321
10.2.2.18 PORTUPD 329
10.2.2.19 SWITCHDATA 329
10.2.3 Assistant Objects - Detailed Description 330
10.2.3.1 ACD_AGENT 330
10.2.3.2 ACD_ANNOUNC 331
10.2.3.3 ACD_GROUP 332
10.2.3.4 ACD_MUSIC 332
10.2.3.5 ACD_NUMBER 333
10.2.3.6 ACD_RCG 333
10.2.3.7 ACD_RCG_SUB 334
10.2.3.8 ACD_RTG_TABLE 334
10.2.3.9 ACD_RTG_TABLE_SUB 335
10.2.3.10 ACD_SHIFT_SET 335
10.2.3.11 ACD_SHIFT_SET_SUB 336
10.2.3.12 ACD_SYS_PARAM 336
10.2.3.13 BOARD 337
10.2.3.14 UBGDAT 348
10.2.3.15 COP 348
10.2.3.16 COT 354
10.2.3.17 FIREMAC 362
10.2.3.18 FIREWALL 363
10.2.3.19 IPROUTING 363
10.2.3.20 NETWORK 364
10.2.3.21 PM_RTG_ACI 365
10.2.3.22 PM_RTG_FAXO 365
10.2.3.23 PM_RTG_MWI 365
10.2.3.24 REFTA 366
10.2.3.25 REMOTE_TERM 367
10.2.3.26 SCREEN_DATA 368
10.2.3.27 SCREEN_PATTERN 368
10.2.3.28 SCREEN_PATTERN_SUB 369
10.2.3.29 TREF_ATTENUATION 369
10.2.3.30 TREF_TONE 369
10.2.3.31 TRKDEV_BOARD 370
10.2.3.32 TRK_ACC_CODE 370
10.2.3.33 TRK_ACC_CODE_SUB 372
10.2.3.34 TRUNK 374
10.2.3.35 TRUNK_GRP 380
10.2.4 HIM Objects - Detailed Description 380
10.2.4.1 BOARD_TYPE 381
10.2.4.2 CABINET 381
10.2.4.3 CABINET_1 382
10.2.4.4 CABINET_VIEW 382
10.2.4.5 CENTRAL_SIU 383
10.2.4.6 CENTRAL_SIU_VIEW 383
10.2.4.7 CONSY_CAB 384
10.2.4.8 CONSY_FRAME 384
10.2.4.9 COS_DESCRIPTION 384
10.2.4.10 CUST_FEATURE 385
10.2.4.11 DBC_CONF_DATA 385
10.2.4.12 DBC_HW_DATA 386
10.2.4.13 DBC_SYS_DATA 387
10.2.4.14 DIMSU_VIEW 388
10.2.4.15 FEA_HEADER 388

10.2.4.16 FEA_SOLD....389
10.2.4.17 FEA_USAGE....389
10.2.4.18 HW_INFO....390
10.2.4.19 LCOS_DESCRIPTION....390
10.2.4.20 LW_ON_CB....391
10.2.4.21 LW_ON_CENTRAL_SIU....391
10.2.4.22 LW_ON_PROCESSOR....392
10.2.4.23 MAC_ADDRESS_VIEW....392
10.2.4.24 PATCH....393
10.2.4.25 PEN_LW....393
10.2.4.26 PORT_STATION_VIEW....394
10.2.4.27 PROJ_PLAN_DATA....394
10.2.4.28 PSIO_VIEW....394
10.2.4.29 REPLACED_AMO....395
10.2.4.30 SCSI_DEV....395
10.2.4.31 SCSI_DEV_VIEW....396
10.2.4.32 SERIAL_LINE....396
10.2.4.33 SHELF....397
10.2.4.34 SHELF VIEW....398
10.2.4.35 SWITCH_APS....398
10.2.4.36 SWITCH_NO....399
10.2.4.37 SWU_BOARD....399
10.2.4.38 SWU_BOARD_VIEW....400
10.2.4.39 SWU_N_DATA....401
10.2.4.40 SWU_PORT_ATTENDANT....401
10.2.4.41 SWU_PORT_SPECIAL....402
10.2.4.42 SWU_T_DATA....402
10.2.4.43 SW_VERSION....403
10.2.4.44 TRUNK_VIEW....403
10.2.4.45 UNIX_SW....404
10.2.4.46 WELCOME_PAGE....404
10.2.4.47 Y_APS....405
10.2.5 Accounting Management Objects - Detailed Description....406
10.2.5.1 AM_VNR_PERS....406
10.2.5.2 AM_VNR_PIN....415
10.2.5.3 AM_VNR_SWITCHDATA....417
10.2.5.4 AMW_COMPORG....423
10.2.5.5 AMW_PERSDATA....423
10.2.5.6 AMW_PINDATA....424
10.2.5.7 AMW_SWITCHDATA....425
10.3 Coded Values....425
10.3.1 General Coded Values (0-10)....426
10.3.2 Call Log....426
10.3.3 Signal....426
10.3.4 Displayanzeige....426
10.3.5 PIN Types (41-43)....426
10.3.6 General Coded Values (44-58)....427
10.3.7 Error states (for action protocol)....427
10.3.8 Time Values....428
10.3.9 Bus Types....428
10.3.10 General Coded Values (79 - 96)....428
10.3.11 Switches 100-149....429
10.3.12 typ,rufnum....429
10.3.13 Switches 150-179....429
10.3.14 Port codec_type....430
10.3.15 Protocol types....430

10.3.16 *Devices 200-249....431
10.3.17 Feature 250-269....432
10.3.18 Feature 270-299....433
10.3.19 Boards 300-349....433
10.3.20 ROLM boards 310-330....434
10.3.21 CMI board....434
10.3.22 Boards for V3.6....434
10.3.23 Boards for DMS3.6....435
10.3.24 Boards for Unity....435
10.3.25 headset,devconst....438
10.3.26 recorder,devconst....438
10.3.27 pmm parameter definitions....439
10.3.28 Server 537-549....439
10.3.29 device types....439
10.3.30 Optiset conference....440
10.3.31 exact device types....440
10.3.32 Addon type....441
10.3.33 Object_type for action_protocol 600-630....441
10.3.34 Operations for action_protocol....442
10.3.35 Object_type for action_protocol for LCR (678-690)....443
10.3.36 Rufnummernplan....444
10.3.37 DIMSU's....444
10.3.38 additional dimsu....445
10.3.39 Unity's dimsu....446
10.3.40 fixed call forwarding....451
10.3.41 key functions (850-899)....452
10.3.42 key functions (920-949)....453
10.3.43 Code values for PORT / net_config....454
10.3.44 pickup_sub, pararng and mastergrp....454
10.3.45 *Coded values for LCR-feature....455
10.3.46 LCR DIMSU's....462
10.3.47 V3.5 lcr, lcr_attribute....462
10.3.48 V3.6 lcr, lcr_attribute....463
10.3.49 CMI exact dev types....464
10.3.50 hopping mode....464
10.3.51 V3.5 Values....465
10.3.52 Device types for Unity....465
10.3.53 *Values for dual_mode_skey....468
10.3.54 Values for dtb_access....468
10.3.55 Values for version in cmi_board....468
10.3.56 Values for ic_data_type....468
10.3.57 V3.5 LCR-Values....468
10.3.58 V3.6 LCR Values....469
10.3.59 lrtg, mfvpuls....469
10.3.60 DIMSU....469
10.3.61 pin,pin_type....470
10.3.62 language,switch/port....471
10.3.63 Vermittlungsplatz Rufnummer....471
10.3.64 Flex-Shelf-Values....472
10.3.65 fix call forward....472
10.3.66 Values for OpenScape Xpressions (Xpressions 450)....472
10.3.67 special_subscriber, line_dist....473
10.3.68 port,classmrk....473
10.3.69 display_name,charcon....473
10.3.70 again Object_type for action_protocol....473
10.3.71 Special values....479

10.3.72 ** section for subdomain codes (also another section is at the end)....479
10.3.73 coded values for stations (unity)....501
10.3.74 dynamic, opt_roomchar....502
10.3.75 switch, upload_txt....502
10.3.76 Devices....503
10.3.77 port, offtype....503
10.3.78 port, pulslev....503
10.3.79 port, pulstype....503
10.3.80 port, ackst....504
10.3.81 port, act....504
10.3.82 port, net_config....504
10.3.83 port, cmind (berlm) and spec....504
10.3.84 port, ins (inbetr)....504
10.3.85 port, alloc....504
10.3.86 port, sztype....505
10.3.87 port, protocol_class....505
10.3.88 cfw, service....505
10.3.89 cfw, itype....506
10.3.90 cfw, variant....506
10.3.91 port, ring_type....506
10.3.92 port, ops_conn_type....506
10.3.93 pin_check, typechkn....506
10.3.94 station_xref, xref_type....507
10.3.95 pin_check, typesum....507
10.3.96 pinclass, trkcdtyp....507
10.3.97 pinclass, dispid....507
10.3.98 pinclass, cotype....507
10.3.99 pinclass, transfer....508
10.3.100 pinclass, pindur....508
10.3.101 pinclass, pintest....508
10.3.102 vbz, src??_dst??....508
10.3.103 vbz, card??_crdr??....509
10.3.104 lcrauth, format....509
10.3.105 huntgrp, disptype....509
10.3.106 key functions....509
10.3.107 berum, cosact....510
10.3.108 vbzgr, vbz_int??....510
10.3.109 vbzgr, vbz_ank??....510
10.3.110 night_opt, night_opt_type....511
10.3.111 spec_subscriber, spec_sub_type....511
10.3.112 port, speed_dial....511
10.3.113 schedule batch, sb_state_txt....512
10.3.114 Coded values for dialplan....512
10.3.115 Codes for subdomain server....528
10.3.116 Codes for port, reg_type....560
10.3.117 Opera Addon type....560
10.4 Error Texts for ODFs with Import Function....561
10.4.1 Error Texts (English)....561
10.4.2 Error Texts (German)....580

Index....608

1 General Information

This appendix provides a description of the XIE-API which includes application programming interface for C++, the webservice interface usable in tools based on WSDL, and the application interface used for import/export. These interfaces allow the exchange of data between OpenScape 4000 and other applications. It is available for both OpenScape4000 Manager and OpenScape 4000 Assistant.

1.1 Scope of Function

The import/export interface (XIE-API) enables you to exchange data between OpenScape 4000 Manager/Assistant and non-OpenScape 4000 applications. These applications may be running on the OpenScape 4000 Manager/Assistant server or on a system connected to the OpenScape 4000 Manager/Assistant server. The import/export interface consists of five interfaces:

- XIE-API Class Library: a C++ program interface available on windows platform; Microsoft Visual C++ 2008 or 2010 is required for using this interface
- XIE-API file interface: data is exchanged via ASCII files (request and response files); import/export-specific processes are responsible for database access
- XIE-API Export Table Client: a graphical user interface on windows platform that provides the export function of XIE-API and the import function for defined set of objects and the possibility to schedule export/import actions via the MS Windows scheduler.
- XIE-API api2hipath tool: a command line tool that provides export function to ASCII file and import function from ASCII file. This tool is available for MS Windows OS and on OpenScape4000 Manager/Assistant.
- XIE-API Web Service Interface: a HTTPS Interface based on a webservice definition file (WSDL). It wraps the XIE-API ASCII-based file interface into webservice methods. It includes also methods designed and used by UC.

The principal procedures for exchanging data via XIE-API are shown in Figure 1.

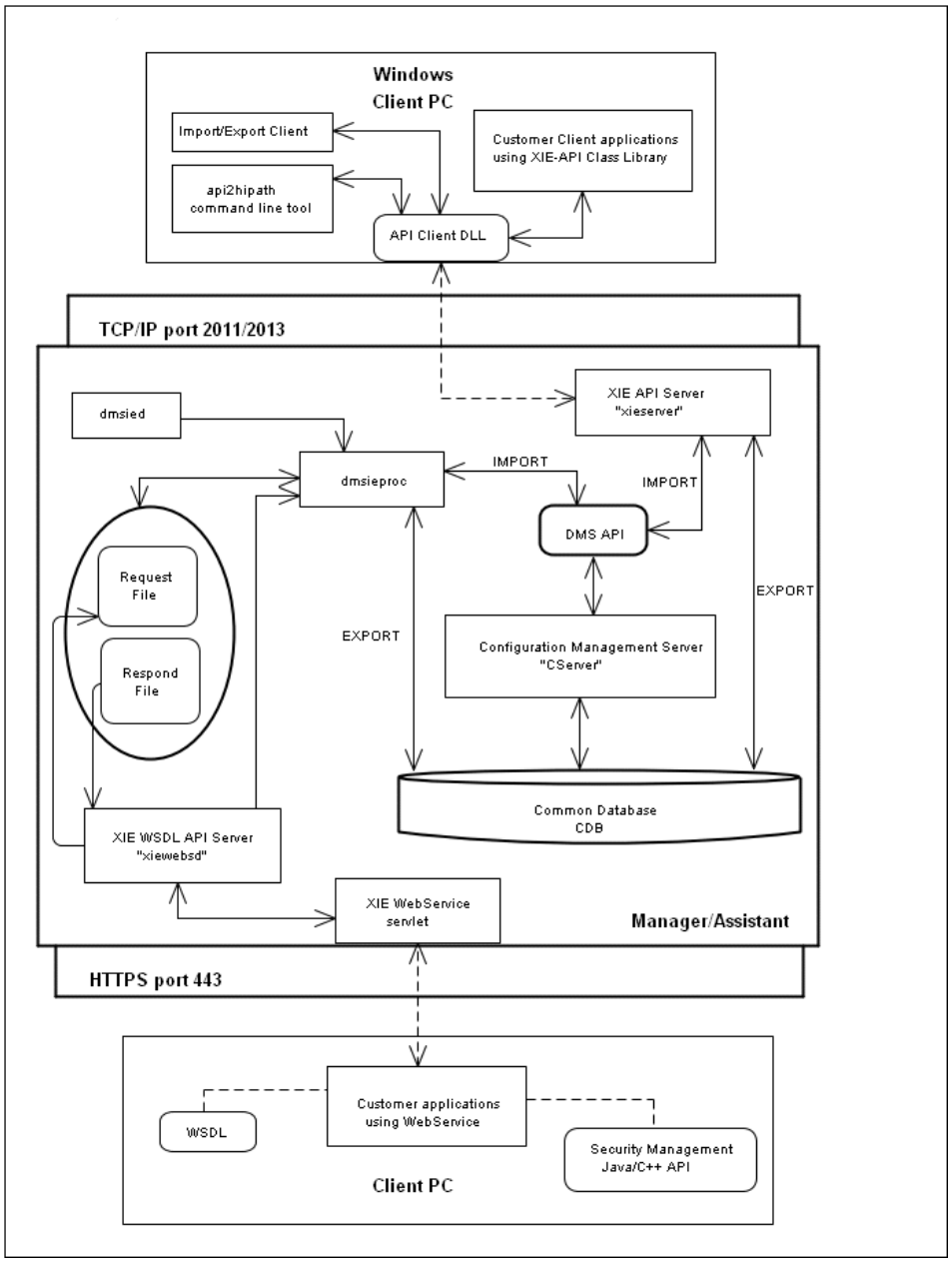

Figure 1: XIE-API Data Exchange

1.2 About This Manual

The objective of this document is to introduce the general concept of XIE-API and the mechanism of writing an application, which will integrate the HiPath/OpenScape Network and OpenScape 4000 Manager into the customer workflow.

Each of the interfaces is described in a separate chapter. General parts, especially the description of the data that can be accessed via XIE-API, but also information for the system administrator, such as installation or the description of server functionality, are separate from the interface description.

Developers who are creating Java or C++ based applications and are familiar with the usage of webservices (WSDL) are suggested to use the XIE-API WebService interface. This interfaces allows them a secure https access into the same list of objects as the XIE-API file interface.

Developers who are creating XIE-API C++ applications and who are familiar with the Windows development environment and do not use WebServices are suggested to use the XIE-API Class Library for their applications. It allows to write complex and interactive applications.

Developers who are not able to use the Windows Visual C++ environment or WebServices can use the api2hipath tool or the file interface, which is independent from special programming techniques or languages.

These interfaces are suitable for writing batch or interactive applications used to synchronize the OpenScape 4000 Manager/Assistant database with any other application.

The Export Table Client is an example of an application written based on the XIE-API Class Library. It serves to import/export data to/from the OpenScape 4000 Manager/Assistant database, therefore its use as base for application programs is restricted compared to the other interfaces. For simplicity, the import is allowed only for a small subset of objects. The benefit of this interface is its simple use. By selecting tables and fields from a list the query to the database is built and the result can be viewed on the screen or exported to a file for further processing. So it can be used also for synchronizing data with the OpenScape 4000 Manager/Assistant database.

1.2.1 System Requirements

Table 1: System Requirements

Server platform	OpenScape 4000 Manager (SLES11)
	OpenScape 4000 Assistant (SLES10, SLES11)
Client platform	Windows 7
	Windows 8
	Windows 10
	Windows 2008
	Windows 2012
	Windows 2016

1.2.2 Hardware Requirements

The OpenScape 4000 Manager/Assistant XIE-API Client software should run on a PC with an Operating System as listed above. The Server Platform may be a OpenScape 4000 Manager or a OpenScape 4000 Assistant.

A TCP/IP connection should provide a connection between the OpenScape 4000 Manager/Assistant and the XIE-API application.

Applications based on the XIE-API Class Lib interface (including XIE-API Client and api2hipath) may choose between:

- WinSockets-based connection without SSL - to server port 2011
- WinSockets-based connection with SSL (uses TLSv1) - to server port 2013

Applications based on the WebService (WSDL) interface use a secure https connection to OpenScape4000 Manager/Assistant. - server port 443.

1.2.3 Software Requirements

The developer should use Microsoft Visual C++ 2008 or 2010 software to create an XIE-API client application.

The XIE-API software package consists of several Visual C++ header files and the XIE-API Visual C++ Class Library.

For the WebService interface, the WSDL can be downloaded directly from the OpenScape 4000 Manager/Assistant. This also applies for

- Java development - SecMCJ.jar library to open a secure session to the OpenScape system
- C++ development - secm.dll library to open a session to the OpenScape system

A WebSerive Client application example written as NetBeens project can also be downloaded.

2 Installation

2.1 Server Installation

For the OpenScape 4000 Assistant/Manager, all packages required for the server side are active and ready to use after the OpenScape 4000 Manager installation.

2.1.1 Server restrictions

During the activation of new software fixes, updates, data backups or logical backups of the CDB unit the XIE-API server-side components might be restarted. This means the active client connections might be dropped and therefore clients have to reconnect after these actions.

2.2 Client Installation

The WSDL file for WebService including a demo example of the client, the installation package of the Export Table Client with the api2hipath command line interface, and the installation package for the Application programmers interface are located in the OpenScape 4000 Manager/Assistant launchpad.

All the files can be downloaded from the launchpad in the "Utilities" section.

If "Utilities" are active, "Import/Export API (XIE)") has to be selected, to get the installation packages for the Export Table Client and the application programmers interface

The following picture shows the location of the installation programs within the launchpad:

Figure 2: Location within the Launchpad

2.2.1 Installing the Programmers Interface (DLL)

When "Install Programmers Interface (DLL)" is selected in the launchpad, the following window appears on the screen:

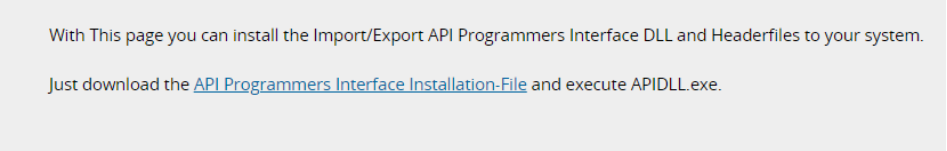

Figure 3: API Programmers Interface Installation

Clicking on the underlined text, the user starts the download of the XIE-API Programmers Interface DLL and headerfiles self-installation package.

Double-clicking "APIDLL.exe" starts the real installation wizard of the XIE-API Programmers interface.

2.2.2 Installing the Export Table Client and api2hipath Tool

When "Install Export Table Client" is selected in the launchpad, the following window appears on the screen:

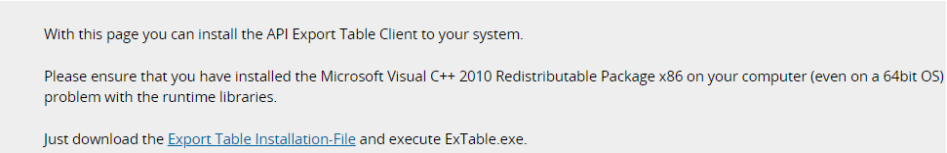

Figure 4: Export Table Client Installation

Clicking on the underlined text, the user starts the download of the Export Table Client and the api2hipath tool self-installation.

Double-clicking "ExTable.exe" starts the real installation of the Export Table Client and api2hipath tool.

3 Server-Side Components

3.1 Directory structure

The import/export files and modules are distributed among different directories, depending on their functions.

By default it is not necessary to touch any of the files located on the server-side file system. However, in some special cases it might be required.

The following directories might be interesting from the user point of view:

- /opt/xie/dmsie/CLaccess - contains server-side configuration files for clients connected via XIE-API class library, api2hipath or Export Table Client
- /opt/xie/dmsie/CLobjects - contains object files accessible for clients connected via XIE-API class library, api2hipath or Export Table Client
- /opt/xie/dmsie/odf - contains object files accessible for clients connected via WebService or clients using XIE file interface
- /opt/xie/dmsie/odf/drv - contains derived object files accessible for clients connected via WebService or clients using XIE file interface
- /opt/xie/dmsie/odf/mpp - contains mapping files used for mapping between XIE object files and Configuration Management data
- /var/xie/trace - directory for creating traces in case of development support is needed

The following files might be interesting from the user point of view:

- /opt/xie/dmsie/usr/xie.ini - configuration file for server-side file interface behaviour
- / opt/xie/dmsie/CLaccess/xieserver.cfg - configuration file for server-side used by Export Table Client, api2hipath and XIE-API class library interface

3.2 File Interface

The file interface basically has the following workflow:

1) Via Security Management (Access Management), the administrator can give XIE-API access rights for the user (see Chapter 4, "Assigning Import/Export API Access Rights via OpenScape 4000 Manager/Assistant Account Management").

2) Once the user has access rights, the main file interface daemon (dmsied) creates an "iefiles" directory in his home folder (this may take several minutes, depending on the api_dsleep setup - see Section 3.2.1, "dmsied")

3) As soon as the /iefiles directory has been created in the user's home folder, he can create ASCII request files (text files with exact syntax - see Section 3.3, "Server side for Export Table client, api2hipath and user programs based on API Class Programmers interface")

4) The Dmsied daemon checks the user"s $HOME/iefiles directories every 15 minutes (dependsing on the api_dsleep setup); if it finds a file with .req suffix, it executes the dmsieproc binary for processing this request file.

5) If the user has access to objects defined in the request file, the dmsieproc binary processes the request and makes sure the respond file is saved in the

same directory. The respond file has to have the same name as the request file with .rsp suffix.

6) If the .req file is successfully processed, it is removed form the user's $HOME/iefiles directory.

3.2.1 dmsied

dmsied is a daemon process which is responsible for processing the request files in the $HOME/iefiles directory, checks the access rights and executes the database requests. It processes only the requests for those users, which have been assigned XIE-API features (ODF"s) via OpenScape 4000 Access Management. dmsied checks by default every 15 minutes if there are new request files in the user"s $HOME/iefiles directory. This default value can be configured by setting the Parameter DSLEEP Variable in Script $XIEHOME/dmsie/bin/dmsied.sh or modifying the api_dsleep field in the CDB-admin table; e.g. to set the time to 5 minutes run following SQL command:

dmsied is a daemon process (located at /opt/xie/dmsie/bin/) responsible for processing the request files in the $HOME/iefiles directory; it checks the access rights and executes the database requests. It processes only the requests for those users which have been assigned XIE-API features. Processing of request files is done each 15 minutes by default.

This can be changed by change of api_dsleep column in admin table of CDB database.

Example: Setup check interval for dmsied to 5 minutes (from server command line):

```
. /opt/Informix/ids_env_var
echo "update admin set api_dsleep=5" | dbaccess cdb
procadmin - t - d dmsied
procadmin - s - d dmsied
```

3.2.2 File Interface Configuration

Apart from the configuration of the dmsied daemon, the administrator can also configure the global behaviour for the processing.of file interface files.

The following parameters can be added/changed in the /opt/xie/dmsie/usr/xie.ini file:

Table 2: File Interface Configuration Values

Parameter	Default value	Description
DELUPD_DAYS	30	expiration period of entries in delta tables
DELRES	48	expiration time of response files in hours
MAXLINE	999	Max. number of lines in XIE-API File Interface Protocol File

Parameter	Default value	Description
DELIMITER	'|'	XIE-API Response file delimiter
NUMBER_OF_OBJECTS	1	Objects per Container

3.2.3 File Interface Protocol File

The file interface logs all requests into the file /opt/xie/dmsie/org/ieprot file.

The logging entries contain the following information:

- name of request file
- Information about the user
- Start date and time of processing the request file
- End date and time of processing the request file

3.3 Server side for Export Table client, api2hipath and user programs based on API Class Programmers interface

The workflow of this interface is as follows

1) Via Security Management (Access Management), the administrator can give XIE-API access rights to the user (see Chapter 4, "Assigning Import/Export API Access Rights via OpenScape 4000 Manager/Assistant Account Management").
2) After step 1, the user can start to use the Export Table client, api2hipath or any other program based on the API Class interface.
3) The user provides his username and password to the client application which connects to the server-side multi-client daemon - the so-called xieserver - via TCP/IP socket stream.
4) Based on the account information, the xieserver refuses or accepts the connection and starts the child process for future data traffic with the client.
5) The user can interactively communicate with the server side via his client program and perform selects/updates/deletes via the Object Data Files for which he has been given access in step 1.

3.3.1 xieserver

The XIE-API Server process "xieserver" is a daemon process that listens for connection requests on the LAN (TCP/IP socket stream). As soon as a client (i.e. XIE-API "Export Table" client) registers a child process is created that handles the further data traffic with this client.

The client can establish and use communication with the xieserver over a non-ssl (2011) or an ssl (2013) port. (This can be selected on the client side and depends on the ports for the administrator has allowed communication through the customer-side firewall).

The child is responsible for:

- Handling the communication with the registered client

- Handling the communication with the Configuration Management processes (via DMSAPI interface; this process actually performs the write database access)
- Performing the read database access (directly via SQL commands) and preparing the results for the client

3.3.1.1 xieserver Configuration

The Following xieserver parameters stored in the file */opt/xie/dmsie/CLaccess/xieserver.cfg* can be changed by the administrator to tune the xieserver daemon:

Table 3: XIE-API Server Configuration Values

Parameter	Default value	Description
MAX_CLIENTS	10	Maximum number of concurrent active XIE-API Server clients
SERVER_TIMEOUT	28800	Number of seconds the XIE-API Server process waits for requests on a active connection before the connection is terminated
TURBOKEY_EVENT	ON	When the flag is set LC-Objects are sent immediately to the Hicom before the connection is terminated
MAX_PROT_LINES	100	Maximum number of lines in the logging file
SELECT_UNIQUE	OFF	OFF ... display all rows, ON ... remove duplicate rows
TENANT	ON	Multitenant support

3.3.1.2 XIE-API Server Log File

The xieserver process logs all attempts of XIE-API client applications to register as an XIE-API client into the file */opt/xie/dmsie/CLaccess/xieprotocol*.

The logging entries contain the following information:

- Date and time
- Activity, setup, close, server shutdown, server start
- Information about the user
- Information about the settings of the environment parameters

3.4 Server Side for WebService Interface

Basically, the server-side WebService interface wraps the above decribed File Interface into a secure form accessible via https.

1) Via Security Management (Access Management), the administrator can give XIE-API access rights for the user (see Chapter 4, "Assigning Import/Export API Access Rights via OpenScape 4000 Manager/Assistant Account Management").
2) Once the user has access rights, he can use any client application written based on the WSDL definition + SecM library to connect to the server and execute export/import requests.
3) The connection is made via https to a tomcat container where servlets are processing the client webservice calls.
4) If the call requires export/import access to the Configuration Database, the servlet forwards the request to the xiewebsd deamon process.
5) The xiewebsd daemon process checks whether the user has access to the Object Data File and if yes, it creates a request file in File Interface format and forwards it to the dmsieproc process for processing.
6) Once the processing is complete, the client can read the content of the respond file via webservice calls.

3.5 Trace/Debug options for server-side components

Each XIE-API daemon or process creates its own trace output file if the traces are activated.

To activate traces, create a file called any.tc in the directory /var/xie/trace with the following content:

```
-o 400 @
-t any
-m 5 !
```

and restart the XIE processes with the following commands:

```
procadmin - t - g XIEAPI
procadmin - s - g XIEAPI
```

To deactivate the traces, remove the *any.tc* file from the directory */var/xie/trace* and restart the XIE processes using the following commands:

```
procadmin - t - g XIEAPI
procadmin - s - g XIEAPI
```

The trace outputs are written in encoded form to the following trace files:

/var/xie/trace/<requestfilename>.td	File Interface trace file
/var/xie/trace/dmsied.td	File Interface Daemon trace file
/var/xie/trace/xie_cl_server.td	xieserver trace file
/var/xie/trace/xie_child_<nr>.td	xieserver child process trace file
/var/xie/trace/xiewebsd.td	WebService xie trace file
/var/log/tomcat6/XIEservice.log	Manager WebService servlet logs
/var/log/tomcat5/base/XIEservice.log	Manager WebService servlet logs

/The content of the traces files is written in binary form. To change it into readable form, use the following command:

```
/opt/CM/sad/bin/trout <requestfilename>.td > my_trace.log
```

3.6 Object Description Files (ODFs)

Object description files describe the OpenScape 4000 Manager/Assistant data from a user point of view. ODFs are the base for any kind of data exchange provided by XIE-API, namely

- the classes XIEQueryInstance, XIECreateInstance, XIEDeleteInstance, XIEModifyInstance, XIEDeleteInstance of the XIE-API Programmers Interface
- request files of the file interface
- display of database tables of the Export Table Client
- request contents of the WebService interface

must refer to an ODF.

ODFs are like some kind of database table from where the values can be read or the values can be updated / inserted; the changes are directly propagated to endpoint devices.

The ODFs are internal OpenScape 4000 Manager/Assistant files created during the installation of the XIE-API package using the database information of Configuration Management. Therefore, there is a correlation between the database table names and the ODF names.

On the server, the created ODFs are stored in the directory */opt/xie/dmsie/odf* (File Interface, WebService Interface) and */opt/xie/dmsie/CLobjects* (XIE-API-Programmers Interface, api2hipath interfaces and Export Table interface).

The created ODFs are readable with any editor and contain detailed information about the fields that are accessible via the special ODF, field names and types (especially the length of the character fields).

The list of Object Data Files and operations which can be executed on them differs based on the interface used.

To see the complete list of ODFs and their access possibilities, see Chapter 10, "Description of ODFs".

3.6.1 Customer Specific Fields

Customer-specific fields which were created using "Dynamic Layout/Customer Specific Field - Tools" can be included in ODFs. For this purpose, the following programs which incorporate the changes in the ODFs are provided:

- ujgeupdodf.sh: updates the File Interface ODFs in the directory */opt/xie/dmsie/odf*
- xiupdodf.sh: updates the DLL XIE-API Interface ODFs in the directory */opt/xie/CLobjects*

Both programs should be called from the */opt/xie/dmsie/bin* directory whenever the user wants to access customer-specific fields via the Export/Import Interface.

3.7 Delta Update Flag in CDB "admin" Table

To obtain the deleted records (used by the SELECT_UPDATES command in the file interface and XIEDeltaQueryInstance API DLL class), the flags port_update, pin_update and pers_update must be set to "001" in the cdb database table "admin".

To change settings use the following commands:

```
/opt/xie/dmsie/bin/ujgedelupd.sh -y (set pin/
port/pers_update in admin table to "001")

/opt/xie/dmsie/bin/ujgedelupd.sh -u (reset pin/
port/pers_update in admin table to "000")
```

ujgedelupd.sh is also responsible for deleting entries in cdb pin_update/ port_update and pers_update tables. A cron-cob table entry starts ujgedelupd.sh every morning at 5:00 a.m. The script deletes every entry which exceeds the expiration period DELUPD_DAYS variable of the /opt/xie/dmsie/ usr/xie.ini file.

4 Assigning Import/Export API Access Rights via OpenScape 4000 Manager/Assistant Account Management

Import/Export API (XIE) security is controlled by OpenScape 4000 Access Management. Object description files (ODFs) are regarded as access rights that can be assigned to or removed from a specific user.

After installation, the users "rsca" and "cust" are assigned all import/export API (XIE) access rights.

To assign access rights (ODFs) to a specific user the following steps must be carried out:

- Select "Account Management" / "Access Right Configuration" via OpenScape 4000 Manager/Assistant launchpad "Access Management" section
- Assign the access right group "All access rights of Import/Export API (XIE)" (ODF names) to a specific user

To assign individual access rights of Import/Export API (for example, only access to PERSDAT and DISPLAY_NAME ODFs) it is possible to create new access right groups via Access Management / Account Management / Access Right Group Configuration.

There are two views in Access Rights Group Configuration: The default view is "Access Rights - Show Component Tree". In this view every feature (ODF) is listed under "Import Export API". In the view "Access Rights - Show Application Tree" the features are separated according to

- API Access Right Configuration - here you can choose which operations are permitted
- Derived Objects for File Interface only - ODFs for special purposes (for AM, DTB, UC)
- OpenScape 4000 Objects - ODFs accessible over both interfaces (File Interface or API Class Interface) from both Manager and Assistant
- Install Export Table Client - access to Export Table Client page from LAP2
- Install Programers Interface DLL - access to Programmers interface installation page from LAP2
- ADP objects - ODFs (so called non-traditional objects) for interface independent select but only via OpenScape 4000 Assistant
- OpenScape/HiPath Inventory Management objects - ODFs for interface independent select but only via OpenScape 4000 Assistant

Switching between these two views is done using the "View" menu and "Show Application Tree" / "Show Component Tree" (see Figure 5).

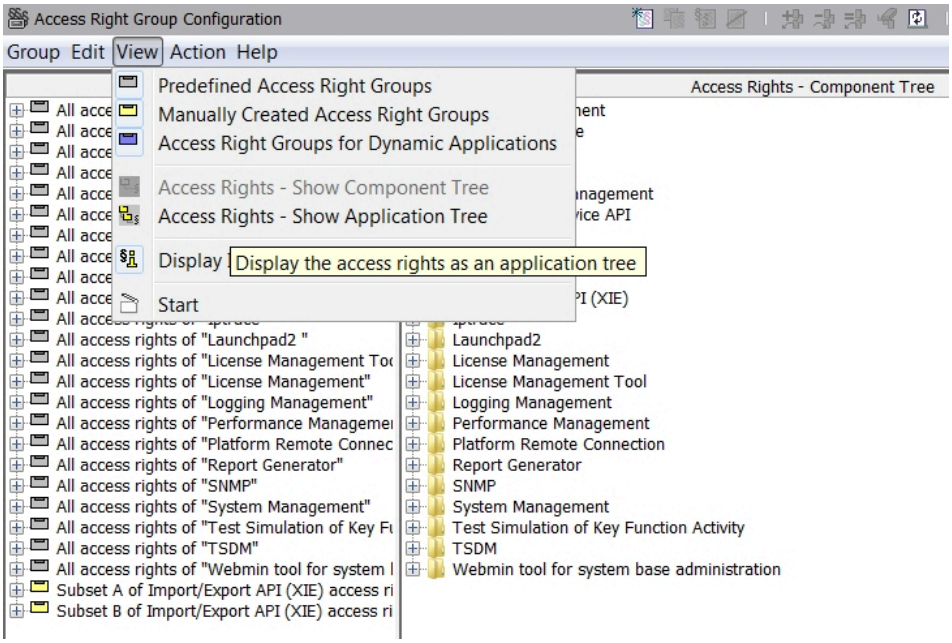

Figure 5: The View Menu in Access Rights Group Configuration

In the following example (see Figure 6), the additional access right groups "Subset A of Import/Export API (XIE) access rights" with the access rights "Install Export Table Client", "PERSDAT" and "DISPLAY_NAME" and "Subset B of Import/Export API (XIE) access rights" with the feature "PERSDAT" were created via "Access Right Group Configuration":

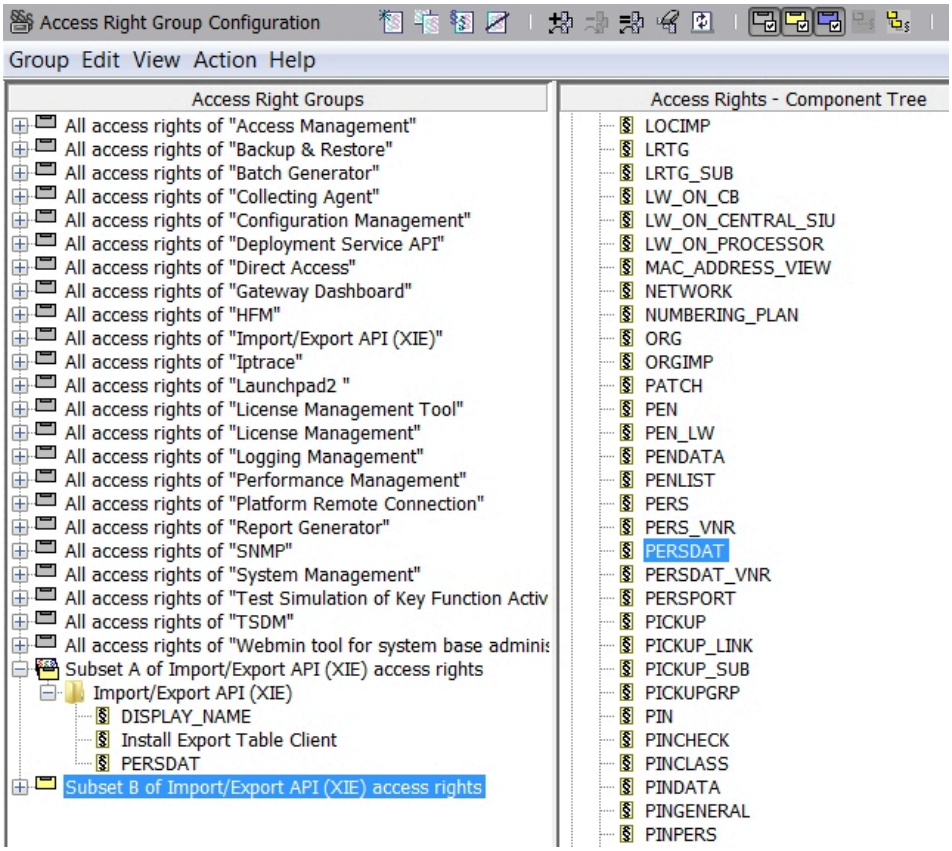

Figure 6: Creating User-specific Access Right Groups

These newly created Access Right Groups are now assigned via Access Right Configuration. As shown in Figure 7, CustomerA has only the rights to Install the Export Table Client and access to the ODFs DISPLAY_NAME and PERSDAT. CustomerB has only access to PERSDAT, and user Boss is assigned all access rights of Import/Export API (XIE).

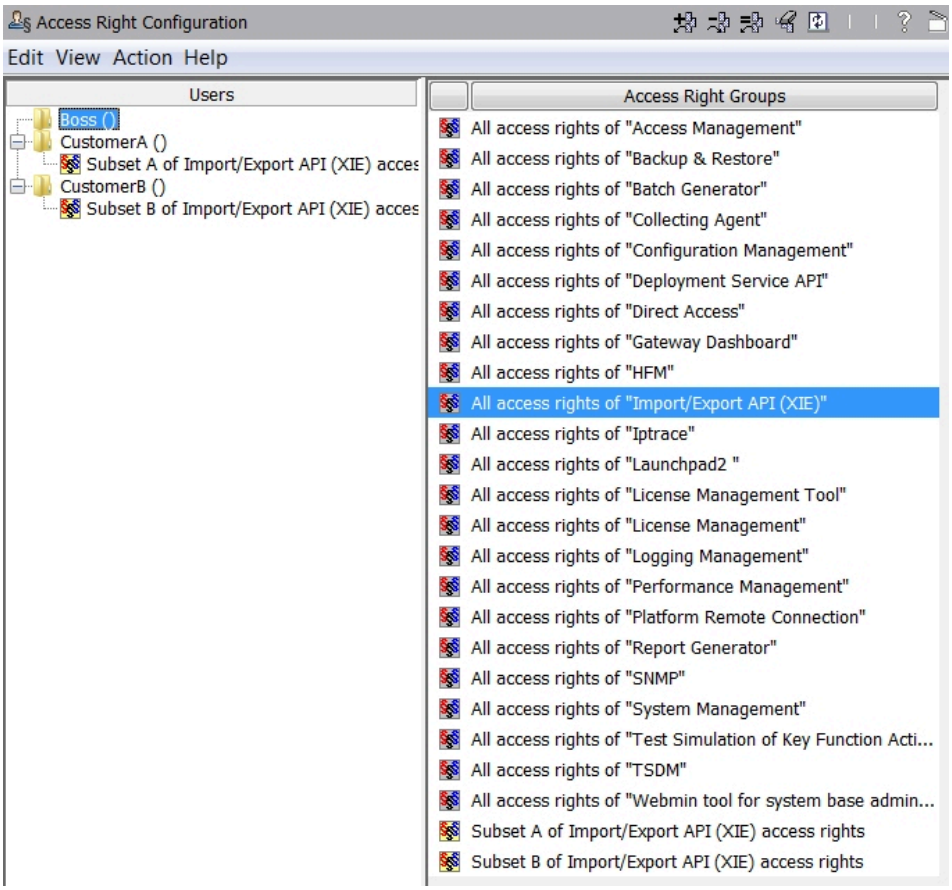

Figure 7: Assigning Access Rights to a Specific User

5 XIE-API Class Library

The OpenScape 4000 Manager/Assistant XIE-API software package provides the server process xieserver and the XIE-API Class Library for the developer, the latter one can be downloaded from the OpenScape 4000 Manager/Assistant server by the use of the launchpad. The developer links the XIE-API Class Library to the application and generates the XIE-API client. Through the TCP/IP connection the customer is connected to the OpenScape 4000 Manager/Assistant server.

5.1 OpenScape 4000 Manager/Assistant XIE-API Class Library Overview

The XIE-API Class Library is the component of the XIE-API that permits the customer to import data into the OpenScape 4000 Manager/Assistant database (cdb) and to export data from OpenScape 4000 Manager/Assistant in a Client/Server environment. In simple words, enhanced XIE-API is a library of function calls, a set of error codes and a standard structure query language syntax for accessing data from a OpenScape 4000 Manager/Assistant database. This way the Enhanced XIE-API provides the most common and convenient methods to access the OpenScape 4000 Manager/Assistant databases for third party applications. Each component of the Enhanced XIE-API provides read access to the OpenScape 4000 Manager/Assistant database. For information about a write access refer to the Section 5.9, "XIE-API Class Library Import Functionality".

5.2 Access to the XIE-API Class Library

XIE-API Class Library does not directly access the OpenScape 4000 Manager/Assistant database and uses the DMSAPI library from Configuration Management to import the data. During the import, DMSAPI verifies that the imported data satisfies certain rules. The rules are different for each object. These rules check data consistency and relations between the attributes. If the check of the data fails then the import is not performed.

The usage of DMSAPI means that import of data via the XIE-API Class Library works the same way as manipulating the data via OpenScape 4000 Manager/Assistant Configuration Management User Interface. There are classes available for opening and closing communication to the OpenScape 4000 Manager/Assistant server, and also for performing database access operations (reading and writing).

5.3 XIE-API Class Library Call

XIE-API Class Library is an MS Visual C++ library. The library has to be linked to the customer"s application so that the application can access OpenScape 4000 Manager/Assistant data as a client.

To build the XIE-API application the customer should call specific classes providing methods for handling the communication between XIE-API client and OpenScape 4000 Manager/Assistant.

5.4 Project Settings

The following XIE-API C++ Class Library should be included into the workspace of the developer"s project:

xieapi.lib	xieapi.lib XIE-API Windows MS-Visual C++ Interface Library to DLL

Includefiles:

xicdelta.H

xiclisel.H

xicommun.H

xicreatx.H

xidbinsx.H

xidefine.H

xidelete.H

xilistex.H

ximodify.H

xixiecom.H

The following DLL file should be located in the application directory and linked to the customer"s application during execution:

xieapi.dll	xieapi.dll XIE-API Windows MS-Visual C++ DLL library

5.5 Run-time Requirements

The customer can install the client application wherever it is convenient for its purposes. The Dynamic Link Library xieapi.dll should be installed and run from the application directory.

The developer can create client applications to run on different Microsoft operating systems (Windows 7, Windows 8, Windows 10, Windows 2008, Windows 2012, Windows 2016). The operating system will map the DLL into the XIE-API process address space when the process is starting up. The process then will execute functions from the DLL. Each application that uses an XIE-API Class Library has its own copy of the library. However, Windows supports multiple applications simultaneously using one copy of the same DLL, therefore the customer can start several different applications running at the same time using the same DLL.

XIE-API Client and OpenScape 4000 Manager server have to be connected through TCP/IP connection (via port 2011 or via port 2013). Port 2013 uses TLSv1.

5.6 XIE-API Class Library Architecture

The XIE-API Class Library represents client/server architecture. The server process will be running on the OpenScape 4000 Manager server as a daemon. The communication between server and client is done by using object "ipcstreams" provided by XIE-API via TCP/IP sockets/winsockets. The communication software is common for both client and server.

5.7 XIE-API Class Library OpenScape 4000 Manager Objects

A set of OpenScape 4000 Manager objects is made available via the XIE-API Class Library. Any instance of the classes for database access refers to exactly one object. The definition of the requested object is done during the creation of the instance by supplying the appropriate parameter value in the constructor. The C++ DBObjType is defined for describing the object.

The potential value of this enumeration type is defined in the header file xixiecom.h

The type DBObjType is defined to describe the available database objects.

The access to the database is offered by providing a set of OpenScape 4000 Manager Objects. These objects describe subscribers, equipment, and functions within the network of OpenScape 4000 Systems. The access rights to these objects are predefined: some provide only export functionality (the customer application may only read the data of this object), other objects also permit import functionality.

XIE-API Objects (ODFs) are the basis for jobs in XIE-API File Interfaces request files and XIE-API C++ Programming Interface.

5.8 XIE-API Class Library Export Functionality

XIE-API Class Library EXPORT functionality enables the customer application to read data from the OpenScape 4000 Manager database.

The export feature searches the database for the specific data. The application defines the object and the required attributes. The application may define also filter criteria and sort criteria. The export functionality delivers only the currently active records from the OpenScape 4000 Manager/Assistant database.

The export feature supports select and select_update operations:

- select: selected data records are fetched from the database.
- select_updates: updated data records and the operations performed on the data records (insert, delete, update) are delivered. This feature supports only PERSDAT, PIN and PORT data.

5.9 XIE-API Class Library Import Functionality

XIE-API Class Library IMPORT functionality enables the customer application to modify data in the OpenScape 4000 Manager database.

- Create operation: new instances of the object are inserted into the OpenScape 4000 Manager/Assistant database (cdb). The data has to satisfy a certain set of rules.
- Delete operation: existing instances of the object are removed from the OpenScape 4000 Manager/Assistant database (cdb).
- Update operation: instances of the objects are modified. The application is described by defining the new values of the attributes that should be updated. The customer can modify more than one instance.

5.10 XIE-API Security

XIE-API Class Library security is controlled by OpenScape 4000 Access Management. The Objects (ODFs) are regarded as access rights that can be assigned to or removed from a specific user. In addition to the Object access rights there are three Access Rights to enable/disable XIE-API Class Library Import functionality:

- API Delete operation allowed
- API Insert operation allowed
- API Update operation allowed

For a description of how to assign/remove access rights, refer to Chapter 4, "Assigning Import/Export API Access Rights via OpenScape 4000 Manager/Assistant Account Management".

5.11 XIE-API Class Library Object Files

The Object Files contain the description of the database objects that are accessible via the XIE-API Class Library.

They contained following information:

- The name and type of accessible attributes of the object
- A list of permitted operations
- The affected database tables

The XIE-API Class Library Object Files are stored only for internal purpose on the OpenScape 4000 server. Any third party application or the system administrator must not modify them. They are stored in the directory /opt/xie/dmsie/CLobjects.

5.12 XIE-API Class Library Key Values

Key Values may be used for limiting the set of records that is affected by a requested database operation. To set a key the application provides the name of the attribute and a value. The value may contain the wildcard characters "*" and "?". Each wildcard character must be preceded by the character "\". The handling of the wildcard character is similar to the SQL commands.

5.13 XIE-API Class Library Comparison Operation

Equality and Inequality are the only supported comparison operations for the key.

Equality is the default assumption whenever a key is provided.

Inequality is requested by supplying a key value that starts with two exclamation marks "!!". The key value "!!OFF" delivers all records where the value of the key attributes is not equal to "OFF".

If there is more than one key provided the single keys are linked by the "AND" operator to define the filter for the operation. The values of all attributes of the record are set as key attributes have to be equal to the supplied key values - then the record satisfies the filter condition.

5.14 Classes

These classes are available as part of the XIE-API Class Library:

- class XIECommunication
- class XIEdbInstance
- class XIEQueryInstance
- class XIEDeltaQueryInstance
- class XIECreateInstance
- class XIEDeleteInstance
- class XIEModifyInstance

The XIE-API Class library provides the following functionality:

- The class "XIECommunication" for handling the communication with the OpenScape 4000 Manager/Assistant. There are methods for opening / closing the communication and retrieving available ODF Names
- The class "XIEQueryInstance" is the class for reading data from the OpenScape 4000 Manager/Assistant Database. There are methods for describing the query (the application may set attributes, keys and order attributes), for performing the query and methods for fetching the records found during the query.
- The class "XIEDeltaQueryInstance " is the class for reading all modifications related to the supplied database object. Only the ODFs PERSDAT, PORT and PIN are supported.
- The class "XIECreateInstance" to import new data into OpenScape 4000 Manager/Assistant. There are methods to set the attribute values and performing the import into OpenScape 4000 Manager/Assistant.
- The class "XIEModifyInstance" to update data in OpenScape 4000 Manager/ Assistant database. Methods are provided for restricting the set of affected records by setting key values and for describing the requested modifications.
- The class "XIEDeleteInstance" is the class for removing records from OpenScape 4000 Manager/Assistant database. Methods are provided for restricting the set of affected records by setting key values. If more than one key is provided all records are deleted that match all the set values.
- In addition to these classes/methods there are additional methods accessible via the "XIEdbInstance" class:
- – method to return information about errors

  - method which delivers information about an ODF (name of available attributes and their data-types)
  - a method to request immediate transmission of command files to the switch (the create/modify or delete operations are performed immediately in the HICOM Network)
  - A method to return the "Action Id". The "Action Id" enables the identification of the operation in the ACTION_CONTROL object.
- All the class-methods are performed synchronously. They return an indication of the outcome of the operation (e.g. the return code OK or some other error code)

5.15 Class XIECommunication

The class provides methods for handling the communication with the OpenScape 4000 Manager/Assistant server. There are methods for opening and closing the connection.

When setting up the connection the user has to provide username, password, server IP address and optionally TLSv1 required usage. The combination username/password has to be assigned to any user via Access Management by the OpenScape 4000 Manager/Assistant administrator.

After having finished the communication with the server the method closeServer() should be invoked to disconnect of the client from the server.

The class XIECommunication is defined in the header file xicommun.H

5.15.1 Class XIECommunication Public Methods

class

XIECommunication() Constructor

~XIECommunication() Destructor

Functions:

int openServer(char* username, char *password, int port, char *hostname, SEC_mode secure_mode) Opens the connection to the server process on the OpenScape 4000 Manager server. The application registers as "username" with "password".

Parameters:

username	input pointer that has to point to the username of the application
password	input pointer that has to point to the password
port	input parameter that describes the port for connection to server (use DEF_PORTNR for SEC_OFF mode and DEF_SEC_PORTNR for SEC_ON secure_mode)

hostname	input pointer that has to point to the IP address of OpenScape 4000 server
secure_mode	Use SEC_OFF for communication without encryption or SEC_ON for communication with TLSv1 encryption

With following possible return values:

OK	connection established
E_CONNECTION_OPEN	connection already open
E_UNKNOWN_USER	user name unknown
E_WRONG_PASSWD	password incorrect
E_CLNT_LIMIT	maximum number of clients exceeded
E_NO_CONNECTION	no connection established
E_TIMEOUT	timeout
E_SEQUENCE	incorrect sequence of Protocol Element
E_MEMORY	memory resource problems

void closeServer() Closes the connection to the OpenScape 4000 Manager/ Assistant server, disconnects the client from the server.

void getObjectNames(char* odfNamesBuffer, int odfNamesLen) Returns a comma separated list of ODF names as currently granted to the user.

Parameters:	
odfNamesBuffer	input pointer that has to point to a memory region allocated by the caller. This memory region will be used to store the retrieved ODF names.
odfNamesLen	input parameter that describes the length of odfNames char buffer; it must be set by the caller to the size of the memory region allocated and provided by odfNamesBuffer.

With following possible return values:

OK	method succeeded
E_SEQUENCE	no previous successful call of openServer()
E_MEMORY	memory resource problems

DBObjType objectNameToId(char* odfName) Returns corresponding DBObjType. This Object Type is used for the definition of the requested object during the creation of the instance by supplying the appropriate parameter value in the constructor.

Parameters:	
odfName	input parameter that points to the name of the ODF

5.15.2 Class XIECommunication Example

```
// XIE_ConsoleApp.cpp : Defines the entry point for the console application.
// Win 95, NT, Win98 Windows 2000 console application.
//Microsoft Visual C++ 6.0

#include "stdafx.h" // include file for MFC include files
#include <iostream> // C++ standard include file

#include "xicommun.h" // XIECommunication class
#include "xiclisel.h" // XIEQueryInstance class

int main(int argc, char* argv[])
{
   int rc; //saves the return codes
   cout << "Starting "<<  endl;
   XIECommunication *commp = new XIECommunication(); //create XIECommunication
    object
   // open communication to server
   // Get user name, password, Port, \\
   rc = commp->openServer("MyName", "MyPasswd", 2011,"225.225.225.0");

   if ( rc ==OK)
   {
      cout<<"Connection OK!"<<endl;
      char objectNames[MAX_OBJECT_LENGTH];
      rc = commp->getObjectNames(objectNames, MAX_OBJECT_LENGTH-1);
      cout<<"available Objects:" << objectNames << endl;
      ..... // do something, function read or write
      commp->closeServer(); // connection is shut down
   }
   else
{
cout<<"No Connection !"<<endl;
   }
   delete commp;
   cout<<"Connection close"<<endl;
   return 0;
}
```

5.16 Class XIEdbInstance

Class XIEdbInstance serves as base class that provides methods for accessing the database. This is an abstract class. The developer cannot create any instances of this class.

The XIEdbInstance class is defined in the header file xidbinsx.H.

5.16.1 Class XIEdbInstance Public Methods

class

XIEdbInstance(XIECommunication *comm) Constructor

~XIEdbInstance() Destructor

Functions:

int getErrorMessage(char *messageBuffer, int messageLength) The methods of the Class Library return information about error in the return code of the methods, where return code is integer value. If more detailed information is required the method getErrorMessage delivers in message "messageBuffer" text information about the error. The method has to be invoked immediately after the detection of an error. If other methods of the XIE-API Class Library have been executed in the meantime the error message is overwritten.

Parameters:	
messageBuffer	input pointer that has to point to a memory region allocated by the caller. This memory region will be used to store the retrieved Error Messages.
messageLength	input parameter that describes the length of messageBuffer char buffer; irt must be set by the caller to the size of the memory region allocated and provided by messageBuffer.

Returns the current error message:

OK	error message received
E_MEMORY	memory resource problems
E_NO_CONNECTION	no connection
E_CONN	problems with connection

int getObjectInfo(char *objInfoBuffer, int objInfoLength) The method delivers information about the current OpenScape 4000 Manager object. On successful return the method retrieves a string that contains all attributes and their types separated by blanks. The attribute-type combinations are separated from each other by the character "|" - example: " attribute1 type1|attribute2 type2|...."

Parameters:

objInfoBuffer	input pointer that has to point to a memory region allocated by the caller. This memory region will be used to store the retrieved object information.
objInfoLength	input parameter that describes the length of objInfoBuffer char buffer; it must be set by the caller to the size of the memory region allocated and provided by objInfoBuffer.

Returns the current error message:

OK	object info received
E_MEMORY	memory resource problems
E_NO_CONNECTION	no connection
E_INFORMIX	Informix error message
E_API	HDMS-API error message
E_XIE	XIE-API Class Library error message
E_CONN	problems with connection

int sendImmediate() Send Command Files immediately to Hicom Network

Returns the current error message:

OK	command send
E_MEMORY	memory resource problems
E_NO_CONNECTION	no connection
E_API	HDMS-API error message

int getActionId() Returns action Ids. The first time getActionId() is used, it returns the first action Id. Each successive time getActionId is used, it returns the next action Id.

Returns the current error message:

positive number	action Id
E_LAST_ID	no more Ids
E_SEQUENCE	no previous call of create(), modify(), or deleteRecord()

virtual void clearAll() = 0;

5.17 Class XIEQueryInstance

XIEQueryInstance is the class for reading data from the OpenScape 4000 Manager. Instances of this class should be created after a successful setup of the communication with the OpenScape 4000 Manager server process.

The XIEQueryInstance class is defined in the header file xiclisel.H.

The set of provided public methods could be split into three groups:

- Methods for describing the query: selAttr(), clearAttr() selAll(). The application may select single attributes or all available attributes for reading access. setKeyAttr(), clearKeyAttr(), setOrderAttr() supplies one or more attributes that define the sorting sequence of the found record. clearAll() clears all settings of attributes.
- Methods for performing the query: query(). As soon as the method for performing the query is invoked the previous queries are regarded to be finished. Records found by the previous query can neither be fetched nor evaluated !
- Methods for fetching the records, found during the query and evaluating them: fetchRecord(), getAttr(), getRecord(). The sequence of the attributes in the found records is determined by the sequence of the setting of the attributes. If single attributes are selected, then the next found record is fetched.

NOTICE: The characters Newline (Hexa 0A) and Pipe ("|") are used internally by Class Library Communication. Therefore these characters are converted internally: Newline is returned as Blank and Pipe is returned as Slash ("/").

5.17.1 Class XIEQueryInstance Public Methods

class

XIEQueryInstance(DBObjType objDesc, XIECommunication *comm_p) Constructor

~XIEQueryInstance() Destructor

The object type of the instance is set and a pointer to the communication object is handed to the instance. The communication to the OpenScape 4000 Manager server has to be open before that function is called.

Parameters:	
objDesc	input parameter that describes the type of the database object
comm_p	input parameter that points to the communication instance

Functions:

int selAttr(char *name) Selects an attribute name for the next query operation.

Parameter:

name	input parameter that points to the name of the attribute

Returns the current error message:

OK	attribute selected
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems
W_ATTR_SET	attribute already set

int selAll() Selects all attributes of the object for the next query operation.

Returns the current error message:

OK	all attributes selected
E_NO_ATTR_SEL	no attribute selected
E_MEMORY	memory resource problems
W_ATTR_SET	attribute already set

int setKeyAttr(char *keyname, char *keyvalue) Sets key attribute as filter attribute for the query.

Parameters:	
keyname	input parameter that points to the name of the key attribute
keyvalue	input parameter that points to the value of the key attribute

Returns the current error message:

OK	key attribute set
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems
W_ATTR_SET	attribute already set

int setKeyAttr(char *keyname, long keyvalue) Sets key attribute as filter attribute for the query.

Parameters:	
keyname	input parameter that points to the name of the key attribute
keyvalue	input parameter that describes the value of the key attribute

Returns the current error message:

OK	key attribute set
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems
W_ATTR_SET	attribute already set

int setAdditionalSql(char *sqlStmt, int uniqueFlag = 0) Sets additional SQL where statement

Parameters:	
sqlStmt	input parameter that points to the statement (SQL syntax, beginning with "and" or "or"; e.g. AND switch_name = "SWI1")
uniqueFlag	input parameter that specifies if duplicate rows should be removed; uniqueFlag = 0 (default) specifies that all rows that satisfy the "WHERE" clause should be displayed. uniqueFlag = 1 specifies that duplicate rows should be removed.

Returns the current error message:

OK	where statement set

int setOrderAttr(char *orderAttr) Sets SQL order attributes.

Parameters:	
orderAttr	input parameter that points to the attribute name for sorting

Returns the current error message:

OK	key attribute set
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems
W_ATTR_SET	attribute already set

int clearKeyAttribute(char *name) Clears key attribute. The attribute name is removed from the set of keys.

Parameters:	
name	input parameter that points to the name of attribute that is removed

Returns the current error message:

OK	key attribute cleared
E_NO_NAME	name of the attribute not supplied

W_ATTR_NOT_SET	attribute not set

int clearAttribute(char *name) Clears attribute. Clears the attribute name that is removed from the set of selected attributes.

Parameters:	
name	input parameter that points to the name of attribute that is removed

Returns the current error message:

OK	attribute cleared
E_NO_NAME	name of the attribute not supplied
W_ATTR_NOT_SET	attribute not set

int query() Performs database access. If no attributes have been set, all attributes are selected. As soon as the method for performing the query is invoked, previous queries are regarded as finished.

Returns the current error message:

return values: >= 0	number of records found
E_NO_ATTR_SEL	no attribute selected
E_MEMORY	memory resource problems
E_CONN	problems with connection
E_NO_CONNECTION	no connection
E_INFORMIX	Informix error message
E_API	HDMS-API error message
E_XIE	XIE-API Class Library error message

int fetchRecord() Fetches found records. Fetches the next found record from the set of records resulting from the previous query(). The method can be invoked only after a previous query call.

Returns the current error message:

OK	record fetched
E_SEQUENCE	missing query call
E_LAST_RECORD	last record has been fetched
E_CONN	problems with connection
E_NO_CONNECTION	no connection
E_MEMORY	memory resource problems
E_INFORMIX	Informix error message

E_API	HDMS-API error message
E_XIE	XIE-API Class Library error message

int getAttr(char *name, char *valueBuffer, int valueLength) Reads attribute from the last found fetched value record.

Parameters:

name	input parameter that describes the name of the required attribute
valueBuffer	input/output parameter that has to point to a memory region allocated by the caller. This memory region will be used to store the retrieved attribute value.
valueLength	input parameter containing the length of valueBuffer char buffer.

Returns the current error message:

OK	attribute received
E_SEQUENCE	missing query call
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems
E_ATTRIBUTE	unknown attribute name

int getRecord(char *foundRecBuffer, int recLength, char delimiter = ‘|’) Delivers complete found record returned by the last fetchRecord() call. The method returns the record as a string. Delimiter separates the values of the single attributes. The sequence is either defined by the sequence of previous selAttr() calls or is equal to the sequence of the attributes in the object description (selAll()).

Parameters:

foundRec	input/output parameter that has to point to a memory region allocated by the caller. This memory region will be used to store the retrieved record.
recLength	input parameter containing the length of foundRec char buffer
delimiter	input parameter that describes the character that separates the single attributes

Returns the current error message:

OK	record received
E_SEQUENCE	missing query call
E_MEMORY	memory resource problems

void clearAll() All settings of the previous key attribute are reset. Erases all previous settings of selected attributes, key attributes and order attributes.

5.17.2 Class XIEQueryInstance Example

```
// XIE_ConsoleApp.cpp : Defines the entry point for the console application.
// Win 95, NT, Win98 Windows 2000 console application.
//Microsoft Visual C++ 6.0

#include "stdafx.h"      // include file for MFC include files
#include <iostream>   // C++ standard include file

#include "xicommun.h" // XIECommunication class
#include "xiclisel.h"      // XIEQueryInstance class

int main(int argc, char* argv[])
{
      int rc; //saves the return codes
      cout << "Starting "<<  endl;

      XIECommunication *commp = new XIECommunication();  //create
       XIECommunication
```

```
object

rc = commp->openServer("MyName", "MyPasswd", DEF_PORTNR,"225.225.225.0");
 if ( rc ==OK)
{
      cout<<"Connection OK!"<<endl;
      // do read and write functions, create query instance

      XIEQueryInstance portQuery(IEPORT,commp); // create query instance
      rc = portQuery.selAttr("extension"); //select attributes
rc = portQuery.selAttr("tcos"); // selected in the query
rc = portQuery.selAttr("ncos"); // selected in the query
rc = portQuery.setKeyAttr("switch_name","MySwitch"); // key is switch_name
rc = portQuery.setOrderAttr("extension"); // supply the order string

int numberOfPortEntries= portQuery.query(); // performing database access
const int MAX_FIELD_LENGTH = 255
char  StationNo[MAX_FIELD_LENGTH];
char  cosDay[MAX_FIELD_LENGTH];
char  cosNight[MAX_FIELD_LENGTH];
for ( i=1; i<= numberOfPortEntries; i++){ // retrieving the values
      rc = portQuery.fetchRecord();
      rc = portQuery.getAttr("extension",stationNo,MAX_FIELD_LENGTH-1);
      rc = portQuery.getAttr("tcos", cosDay, MAX_FIELD_LENGTH-1); //
       retrieve
      cos_day
      rc = portQuery.getAttr("ncos", cosNight, MAX_FIELD_LENGTH-1);//
       retrieve
      cos_night
      cout << "station_no: " << StationNo << " COS day: " << cosDay << "
       COSnight:
```

```
            " << cosNight   << endl;
            }
            commp->closeServer(); // connection is shut down
      }
      else
{
      cout<<"No Connection!"<< endl;
}
      ...// shut down the connection

delete commp;
cout<<"Connection close"<<endl;
return 0;
}
```

5.18 Class XIECreateInstance

XIECreateInstance is the class for importing data into the OpenScape 4000 Manager/Assistant database. Instances of this class should only be created after a successful setup of the communication with the OpenScape 4000 Manager/Assistant server process.

The requested ODF is set as parameter in the constructor. The handling of the attributes is covered by the methods setAttr (sets the value of one attribute), clearAttribute (no value is assigned to the attribute anymore), and clearAll (removes all previous settings of values). These methods set the values only internally. Invoking create causes the actual database access.

The class is defined in the header file xicreatx.h.

NOTICE: The characters Newline (Hexa 0A) and Pipe ("|") are used internally by Class Library Communication. Therefore these characters are converted internally: Newline is stored as Blank and Pipe is stored as Slash ("/").

5.18.1 Class XIECreateInstance Public Methods

Class

XIECreateInstance(DBObjType objDesc, XIECommunication *comm_p)
Constructor

~XIECreateInstance() Destructor

The object type of the instance is set and a pointer to the communication object is handed to the instance. The communication to the server has to be open before that function is called.

Parameter:	
objDesc	input parameter that describes the type of the database object
comm_p	input parameter that has to point to the communication instance

Functions:

int setAttr(char *name, char *value) Sets the attribute name to value value.

Parameters:	
name	input parameter that has to point to the name of the attribute
value	input parameter that has to point to the value of the attribute

1) Return values:

OK	attribute set
W_ATTR_SET	attribute has been set previously
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems

int setAttr(char *name, long value) Sets the attribute name to value value.

Parameters:	
name	input parameter that has to point to the name of the attribute
value	input parameter with the value of the attribute

Return values:

OK	attribute set
W_ATTR_SET	attribute has been set previously
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems

int clearAttr(char *name) No value is assigned to the attribute name anymore.

Parameter:

name	input parameter that has to point to the name of the attribute

Returns the current error message:

OK	attribute removed
W_ATTR_NOT_SET	attribute value was not set
E_NO_NAME	name of the attribute not supplied

int create(char * orderId) Imports the record into the OpenScape 4000 Manager/Assistant database. The return code informs only about the access to OpenScape 4000 Manager/Assistant database but not about the results from the OpenScape 4000 systems in the network.

Parameter:

OrderId	input parameter that has to point to the "Job Number"; string containing up to 12 characters. This "Job Number" enables the identification of the create operation in the ACTION_CONTROL object.

Returns the current error message:

1	record inserted
E_CONN	problems on the connection line
E_NO_CONNECTION	no connection to OpenScape 4000 Manager server
E_ATTRIBUTE	one of the set attributes is unknown
E_NO_PERMISSION	user has no access rights for this object
E_OPERATION	create operation not permitted
E_DB_SERVER	no connection to cserver
E_API	HDMS-API error
E_MEMORY	memory resource problems

int clearAll() All settings of the previous key attribute are reset.

5.18.2 Class XIECreateInstance Example

```
// XIE_ConsoleApp.cpp : Defines the entry point for the console application.
// Win 95, NT, Win98 Windows 2000 console application.
// Microsoft Visual C++ 6.0

#include "stdafx.h"      // include file for MFC include files
#include <iostream>   // C++ standard include file

#include "xicommun.h" // XIECommunication class
#include "xiclisel.h"       // XIEQueryInstance class
#include "xicreatx.h"     // XIECreateInstance

int main(int argc, char* argv[])
{
    int rc; //saves the return codes
    cout << "Starting "<<  endl;
    XIECommunication *commp = new XIECommunication();  //create XIECommunication
     object
    rc = commp->openServer("MyName","MyPasswd", DEF_PORTNR, "ipAddress");  //
     open
    communication to server
    XIECreateInstance  portCreate(IEPORT, commp);//create PORT object
    rc = portQuery.setAttr("switch_name","MySwitchName");// the attributes
    switch_name,domain
    rc = portQuery.setAttr("domain","MyDomain");// names ... are set
    ... //set other attributes
    rc = portQuery.create() // perform the creation
    if (rc != 1) {
    ...// evaluate return code
    }
    commp->closeServer();// close connection
    delete commp;
    return 0;
}
```

5.19 Class XIEDeleteInstance

XIEDeleteInstance is the class for removing records from the OpenScape 4000 Manager/Assistant database. Instances of this class should only be created after a successful setup of the communication with the OpenScape 4000 Manager/Assistant server process. The requested ODF is set as parameter in the constructor. The records that should be deleted are described by key values. The application sets a value for certain attributes. All records where the value of this attribute is equal to the set value are affected by the delete operation. If more than one key value is provided all records are deleted that match all the set values.

The class is defined in the header file xidelete.H.

5.19.1 Class XIEDeleteInstance Public Methods

class

XIEDeleteInstance(DBObjType objDesc, XIECommunication *comm_p) Constructor

~XIEDeleteInstance() Destructor

The object type of the instance is set and a pointer to the communication object is handed to the instance. The communication to the server has to be open before that function is called.

Parameters:	
objDesc	input parameter that describes the type of the database object
comm_p	input parameter that has to point to the communication instance

Functions:

int setKeyAttr(char *keyname, char *keyvalue) Sets the value of an attribute that serves as a key for the delete operation. The attribute keyname is set to value keyvalue.

Parameters:	
keyname	input parameter that has to point to the name of the key attribute
keyvalue	input parameter that has to point to the value of the key attribute

Returns the current error message:

OK	key attribute set
W_ATTR_SET	attribute has been set previously
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems

int setKeyAttr(char *keyname, long keyvalue) Sets the value of an attribute that serves as a key for the delete operation. The attribute keyname is set to value keyvalue.

Parameters:	
keyname	input parameter that has to point to the name of the key attribute
keyvalue	input parameter that has to point to the value of the key attribute

Return values:

OK	key attribute set
W_ATTR_SET	attribute has been set previously
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems

int clearKeyAttr(char *name) No key value is assigned to the attribute name anymore.

Parameters:	
name	input parameter that has to point to the name of the attribute

Returns the current error message:

OK	attribute removed
W_ATTR_NOT_SET	attributes value was not set
E_NO_NAME	name of the attribute not supplied

int setAdditionalSql(char *sqlStmt) Sets additional SQL where statement

Parameters:	
sqlStmt	input parameter that points to the statement (SQL syntax, beginning with "and" or "or"; e.g. AND switch_name = "SWI1")

Returns the current error message:

OK	where statement set

int deleteRecord(char * orderId) Deletes all records from the OpenScape 4000 Manager/Assistant database that match the set key attributes.

Parameters:

OrderId	input parameter that has to point to the "Job Number";string containing up to 12 characters. This "Job Number" enables the identification of the delete operation in the ACTION_CONTROL object.

Returns the current error message:

Positive number	number of deleted records
0	no record deleted
E_CONN	problems on the connection line
E_NO_CONNECTION	no connection to OpenScape 4000 Manager server
E_ATTRIBUTE	one of the set attributes is unknown
E_NO_PERMISSION	user has no access rights for this object
E_OPERATION	delete operation not permitted
E_DB_SERVER	no connection to cserver
E_API	HDMS-API error
E_MEMORY	memory resource problems

int clearAll() All settings of the previous key attribute are reset.

5.19.2 Class XIEDeleteInstance Example

```
// XIE_ConsoleApp.cpp : Defines the entry point for the console application.
// Win 95, NT, Win98 Windows 2000 console application.
//Microsoft Visual C++ 6.0

#include "stdafx.h"      // include file for MFC include files
#include <iostream>   // C++ standard include file

#include "xicommun.h" // XIECommunication class
#include "xiclisel.h"       // XIEQueryInstance class
#include  "xidelete.H"   //  XIEDeleteInstance

int main(int argc, char* argv[])
{
   int rc; //saves the return codes

   cout << "Starting "<<  endl;

   XIECommunication *commp = new XIECommunication();  //create XIECommunication
    object

   rc = commp->openServer("MyName", "MyPasswd", DEF_PORTNR,"225.225.225.0"); //
    open
```

```
   communication to server
   XIEDeleteInstance   personRem(IEPERSDAT, commp) // create Delete  on PERSON
   rc = personRem.setKeyAttr("name", "Maxell"); // defining the keys
   rc = personRem.setKeyAttr("christianname", "John E.");
   rc = personRem.setKeyAttr("company", "MyCompanyName");
   rc = personRem.setKeyAttr("org1", "Marketing");
   rc = personRem.setKeyAttr("switch_name", "ANL2");
   rc = personRem.deleteRecord(); // deleting the record
   if (rc == 1)
        cout << "Record deleted" << endl;
   else
   {
   cout << "Error during deletion: " << rc << endl
   char errMsg[255];
   personRem.getErrorMessage(errMsg, 254);   // get error message
    cout << "Error during deletion: " << errMsg << endl;
   }
   ...  // close communication
   commp->closeServer(); // close connection
   delete commp;
   return 0;
}
```

5.20 Class XIEModifyInstance

XIEModifyInstance is the class for updating records in the OpenScape 4000 Manager/Assistant database. Instances of this class should only be created after a successful setup of the communication with the OpenScape 4000 Manager/Assistant server process.

The requested ODF is set as parameter in the constructor. Methods are provided for restricting the set of affected records by setting key values and for describing the requested modifications. Both are done by assigning values to attributes.

The set of the provided public methods can be split into three groups:

- Methods for setting and unsetting the modify: The setting (and unsetting) of key values is handled by the methods setKeyAttr(), clearAttr() and clearAll(). clearAll() also affects the handling of the attributes that are to be modified.
- Methods for updating: The methods updAttr() and clearAttr() handle the set of attributes to be updated.

- Method for modifying the record. The set of affected records is the set of all records that match the set key values and the old values that are supplied when invoking modify().

The class is defined in the header file ximodify.H.

NOTICE: The characters Newline (Hexa 0A) and Pipe ("|") are used internally by Class Library Communication. Therefore these characters are converted internally: Newline is stored as Blank and Pipe is stored as Slash ("/").

5.20.1 Class XIEModifyInstance Public Methods

Class

XIEModifyInstance(DBObjType objDesc, XIECommunication *comm_p) Constructor

~XIEModifyInstance() Destructor

The object type of the instance is set and a pointer to the communication object is handed to the instance. The communication to the server has to be open before that function is called.

Parameters:	
objDesc	input parameter that describes the type of the database object
comm_p	input parameter that has to point to the communication instance

Functions:

int setKeyAttr(char *keyname, char *keyvalue) Sets the value of an attribute that serves as key for the modify operation. The attribute keyname is set to value keyvalue.

Parameters:	
keyname	input parameter that has to point to the name of the key attribute
keyvalue	input parameter that has to point to the value of the key attribute

Return values:

OK	key attribute set
W_ATTR_SET	attribute has been set previously
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems

int setKeyAttr(char *keyname, long keyvalue) Sets the value of an attribute that serves as a key for the modify operation. The attribute keyname is set to value keyvalue.

Parameter:	
keyname	input parameter that has to point to the name of the key attribute
keyvalue	input parameter that has to point to the value of the key attribute

Return values:

OK	key attribute set
W_ATTR_SET	attribute has been set previously
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems

int updAttr(char *name, char *oldValue, char *newValue) Marks the attribute name for modification. If the current value of the attribute is oldValue, the value of the attribute is changed to newValue.

Parameters:	
name	input parameter that has to point to the name of the attribute
oldValue	input parameter that has to point to the previous value of the attribute; supply an empty string if the old value is unknown.
newValue	input parameter that has to point to the new value of the attribute

Return values:

OK	key attribute set
W_ATTR_SET	attribute has been set previously
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems

int updAttr(char *name, long oldValue, long newValue) Marks the attribute name for modification. If the current value of the attribute is oldValue the value of the attribute is changed to newValue.

Parameters:	
name	input parameter that has to point to the name of the attribute

oldValue	input parameter that has to point to the previous value of the attribute; supply an empty string if the old value is unknown.
newValue	input parameter that has to point to the new value of the attribute

Return values:

OK	key attribute set
W_ATTR_SET	attribute has been set previously
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems

int clearAttr(char *name) The previous setting of the attribute name (either as key or for a modification) is reset.

Parameter:

name	input parameter that has to point to the name of the attribute

Return values:

OK	attribute removed
W_ATTR_NOT_SET	attributes value was not set
E_NO_NAME	name of the attribute not supplied

int setAdditionalSql(char *sqlStmt) Sets additional SQL where statement

Parameters:

sqlStmt	input parameter that points to the statement (SQL syntax, beginning with "and" or "or"; e.g. AND switch_name = "SWI1")

Returns the current error message:

OK	where statement set

int modify(char *orderId) Updates the records from the OpenScape 4000 Manager/Assistant database that match the previously given conditions

Parameter:

OrderId	input parameter that has to point to the "Job Number"; string containing up to 12 characters. This "Job Number" enables the identification of the modify operation in the ACTION_CONTROL object.

Return values:

Positive number	number of modified records
0	no record satisfied these conditions
E_CONN	problems on the connection line
E_NO_CONNECTION	no connection to OpenScape 4000 Manager server
E_ATTRIBUTE	one of the set attributes is unknown
E_NO_PERMISSION	user has no access rights for this object
E_OPERATION	modify operation not permitted
E_DB_SERVER	no connection to cserver
E_API	HDMS-API error
E_MEMORY	memory resource problems

int clearAll() All settings of the previous key attribute are reset.

5.20.2 Class XIEModifyInstance Example

```
// XIE_ConsoleApp.cpp : Defines the entry point for the console application.
// Win 95, NT, Win98 Windows 2000 console application.
//Microsoft Visual C++ 6.0

#include "stdafx.h"      // include file for MFC include files
#include <iostream>   // C++ standard include file

#include "xicommun.h" // XIECommunication class
#include "xiclisel.h"      // XIEQueryInstance class
#include "ximodify.h"  //  XIEModifyInstance

int main(int argc, char* argv[])
{
    int rc; //saves the return codes


    cout << "Starting "<<  endl;


    XIECommunication *commp = new XIECommunication();  //create XIECommunication
     object


    rc = commp->openServer("MyName", "MyPasswd", DEF_PORTNR,"225.225.225.0"); //
     open
    communication to server


    XIEModifyInstance    dispUpdate(IEDISPLAY_NAME, commp);// modify DISPLAY_NAME
     object
    rc = dispUpdate.setKeyAttr("switch_name", "MySwitch"); //describing the
     affected
    record
    rc = dispUpdate.setKeyAttr("extension", "5555"); // (set keys on switch and
    extension)
    rc = dispUpdate.updAttr("name", "old name", "new name"); //describing the
     requested
    mod.
    rc = dispUpdate.modify(); // performing the modification
    if (rc == 1
          cout << "Modification made" << endl;
    else
```

```
    {
          char errMsg [255];
          dispUpdate.getErrorMessage(errMsg, 254);
          cout << "Error during modification: " << errMsg << endl;
    }
          ... // close communication
    commp->closeServer();
    delete commp;  // delete commp

    cout<<"Connection close"<<endl; / /connection closed
    return 0;
}
```

5.21 Class XIEDeltaQueryInstance

XIEDeltaQueryInstance is the class for reading all modifications related to the OpenScape 4000 Manager/Assistant database object.

Only the ODFs PERSDAT, PORT and PIN are supported. To obtain the deleted data records, the flags for port_update, pin_update and pers_update must be set to 001 in the admin table (refer to Section 3.7 on page 3-32).

Instances of this class should only be created after a successful setup of the communication with the OpenScape 4000 Manager/Assistant server process. The application may select single attributes or all available attributes. The application may define filters for the query and define attributes, which define the sorting of the found records.

The set of provided public methods could be split into three groups:

- Methods for describing the query: selAttr(), selAll(), setTime(), clearAttribute(). The application may select single attributes or all available attributes for the reading access. The usage of the methods selAttr and selAll within in the description of one query are contradictory. Therefore if the selection of single attributes and of all attributes is mixed the last called method is relevant. If selAll was the last invoked method all attributes are selected. If the last invoked method was selAttr only the selected attributes are queried.
- Methods for performing the query: deltaQuery(). As soon as the method for performing the query is invoked, previous queries are regarded to be finished.
- Methods for fetching the records found during the query and evaluating them: fetchRecord(), getAttr(), getRecord(). The sequence of the attributes in the found records is determined by the sequence of the setting of the attributes (if single attributes are selected) or matches the sequence in the object definition (see below). As soon as any method for describing the query is invoked previous queries are regarded to be finished. Records found by the previous query can neither be fetched or evaluated.

The class is defined in the header file xicdelta.h

5.21.1 Class XIEDeltaQueryInstance Public Methods

class

XIEQueryInstance(DBObjType objDesc, XIECommunication*comm_p) Constructor

~XIEQueryInstance() Destructor

Function:

The object type of the instance is set and a pointer to the communication object is handed to the instance. The communication to the server has to be open before that function is called.

Parameters:	
objDesc	input parameter that describes the type of the database object
comm_p	input parameter that has to point to the communication instance

int selAttr(char *name) Selects the attribute name for the next query operation.

Parameter:	
name	input parameter that has to point to the name of the attribute

Return the current error message:

OK	selection accepted
W_ATTR_SET	attribute is already set
E_NO_NAME	name of the attribute not supplied
E_MEMORY	memory resource problems

int setTime(char *from, char *to) Sets an Informix time interval for the delta query operation. Specify the from/to times in the Informix date format (YYYY-MM-DD hh:mm:ss).

Parameter:	
from	input parameter that has to point to the start time of the time interval
to	input parameter that has to point to the end time of the time interval

Returns the current error message:

OK	time set
E_FORMAT	wrong timestamp format
E_MEMORY	memory resource problems

W_TIME_SET	time has been previously set

int selAll() Selects all attributes of the object for the next delta query operation. Previous selections of single attributes are erased.

Returns the current error message:

OK	selection accepted
E_NO_ATTR_SEL	no attribute selected
E_MEMORY	memory resource problems
W_ATTR_SET	single attributes have been selected previously

int setAdditionalSql(char *sqlStmt, int uniqueFlag = 0) Sets additional SQL where statement

Parameters:	
sqlStmt	input parameter that points to the statement (SQL syntax, beginning with "and" or "or"; e.g. AND switch_name = "SWI1")
uniqueFlag	input parameter that specifies if duplicate rows should be removed: uniqueFlag = 0 (default) specifies that all rows that satisfy the "WHERE" clause should be displayed; uniqueFlag = 1 specifies that duplicate rows should be removed.

Returns the current error message:

OK	where statement set

int clearAttribute(char* name) The attribute name is removed from the set of selected attributes.

Parameter:	
Name	input parameter that has to point to the attribute that is removed

Returns the current error message:

OK	attribute removed
E_NO_NAME	name of the attribute not supplied
W_ATTR_NOT_SET	attribute was not set

int deltaQuery() Performs the delta query reading access. As soon as the method for performing the query is invoked, previous queries are regarded as finished.

Returns the current error message:

nonnegative number	number of found records
E_CONN	connection problems
E_NO_CONNECTION	no connection to xieserver
E_NO_ATTR_SEL	no attributes selected
E_MEMORY	memory resource problems
E_API	HDMS-API error message
E_XIE	XIE-API Class Library error
E_INFORMIX	INFORMIX error

int fetchRecord() Fetches the next found record from the set of records resulting from a previous deltaQuery() call. The method must be invoked only after a previous query call.

Returns the current error message:

OK	record has been fetched
E_MEMORY	memory resource problems
E_LAST_RECORD	last record has been fetched previously
E_SEQUENCE	no previous successful deltaQuery() call
E_CONN	connection problems
E_API	HDMS-API error message
E_XIE	XIE-API Class Library error
E_INFORMIX	INFORMIX error

int getOperation(opType&updOperation) Returns the operation belonging to the fetched record, if was inserted. OpType are opDelete, opModify and opCreate types.

Parameter:

updOperation	input/output parameter that has to point to the opType. This memory region will be used to store the retrieved operation type (opCreate, opDelete or opModify)

Returns the current error message:

OK	operation received
E_SEQUENCE	missing query call

iint getAttr(char *name, char *valueBuffer, int valueLength) Returns the value of the attribute name in the last fetched record. The method must be called only after previous calls of deltaQuery() and fetchRecord(). Each datatype is returned as a string value.

Parameter:	
name	input parameter that has to point to the name of the required attribute
valueBuffer	input/output parameter that has to point to a memory region allocated by the caller. This memory region will be used to store the retrieved attribute value.
valueLength	input parameter that has to be set to the size of the memory region allocated and provided by valueBuffer.

Returns the current error message:

OK	attribute delivered
E_ATTRIBUTE	attribute has not been selected
E_NO_NAME	no value supplied for name
E_MEMORY	memory resource problems
E_SEQUENCE	no previous successful call of fetchRecord()

int getRecord(char *foundRec, int recLen, char delimiter = "|") Delivers the record returned by the last fetchRecord call. The method returns the record as a string. Delimiter separates the values of the single attributes. The sequence is either defined by the sequence of previous selAttr() calls or is equal to the sequence of the attributes in the ODF selAll().

Parameter:	
foundRec	input/output parameter that has to point to a memory region allocated by the caller. This memory region will be used to store the retrieved record.
recLen	input parameter that contains the length of foundRec char buffer
delimiter	input parameter that describes the character that separates the single attributes

Returns the current error message:

OK	record delivered
E_MEMORY	memory resource problems
E_SEQUENCE	no previous call of fetchRecord()

5.21.2 Class XIEDeltaQueryInstance Example

```
// XIE_ConsoleApp.cpp : Defines the entry point for the console application.
// Win 95, NT, Win98 Windows 2000 console application.
//Microsoft Visual C++ 6.0

#include "stdafx.h"      // include file for MFC include files
#include <iostream>   // C++ standard include file

#include "xicommun.h" // XIECommunication class
#include "xiclisel.h"      // XIEQueryInstance class
#include "xicdelta.h"     // XIEDeltaQueryInstance

int main(int argc, char* argv[])
{
    int rc; //saves the return codes
    cout << "Starting "<<  endl;
    XIECommunication *commp = new XIECommunication();  //create XIECommunication
     object
    rc = commp->openServer("MyName", "MyPasswd", DEF_PORTNR,"225.225.225.0"); //
     open
```

```
    communication to server
    XIEDeltaQueryInstance deltaQuery(IEPERSDAT, commp); // create PERSDAT Delta
     Query

    rc = deltaQuery.setTime("2000-01-01 00:00:00", "2001-12-31 11:00:00");
    rc = deltaQuery.selAttr("extension"); // select field station_no
    rc = deltaQuery.selAttr("name"); // select field name
    rc = deltaQuery.selAttr("unique_key"); // select field unique_key
    rc = deltaQuery.selAttr("modify_date"); // select field modify_date
    int numberOfEntries = deltaQuery.deltaQuery();  // performing the deltaquery
        // numberOfEntries contains the number of found records
    if (numberOfEntries < 0)   // retrieve the error message
    {
        char errMsg[255];
        deltaQuery.getErrorMessage(errorMsg, 254);
        cout<<"Error during quering"<<errMsg<<endl;
    }
els
```

```
e
{
    // reserve space for attributes:
    const int MAX_REC_LENGTH = 255;
    char station_no[MAX_FIELD_LENGTH];
    char name[MAX_FIELD_LENGTH];
    char unique_key[MAX_FIELD_LENGTH];
    char modify_date[MAX_FIELD_LENGTH];
    opType operation;
    // read all records in a loop ...
    for (i = 0; i < numberOfEntries; i++)
    {
        rc = deltaQuery.fetchRecord();// fetch the next record
        rc = deltaQuery.getOperation(operation);
        // read the operation belonging to the record
        switch(operation)
        {
            case opCreate: cout << "Record was inserted " << endl; break;
            case opDelete: cout << "Record was deleted  " << endl; break;
            case opModify: cout << "Record was updated  "  << endl; break;
        }
        // read the attribute values:
        rc = deltaQuery.getAttr("extension", station_no, MAX_FIELD_LENGTH -1);
        rc = deltaQuery.getAttr("name", name, MAX_FIELD_LENGTH -1);
        rc = deltaQuery.getAttr("unique_key", unique_key, MAX_FIELD_LENGTH -1);
        rc = deltaQuery.getAttr("modify_date", modify_date, MAX_FIELD_LENGTH
         -1);
        cout << "Extension:" << station_no << " Name:" << name << endl;
        ...
    }
}
// close communication
commp->closeServer();
delete commp;  // delete commp
cout<<"Connection close"<<endl; //connection closed
return 0;
}
```

6 File Interface

The file interface provides a Client/server solution that enables partner applications to exchange data with OpenScape 4000 Manager/Assistant via ASCII files.

6.1 Basic Concepts

The most important components for exchanging data via ASCII files are listed below:

- `Object description files (ODFs):` Describe the data and type of access.
- `Request and response files:` Request files are requests for reading or writing data; response files contain the results.
- `dmsieproc`: Processes requests in a request file and generates a response file based on the results of the orders completed.
- `dmsied:` Daemon process which controls processing of the request files.

Figure 8 shows how the five components listed above interact during a data exchange:

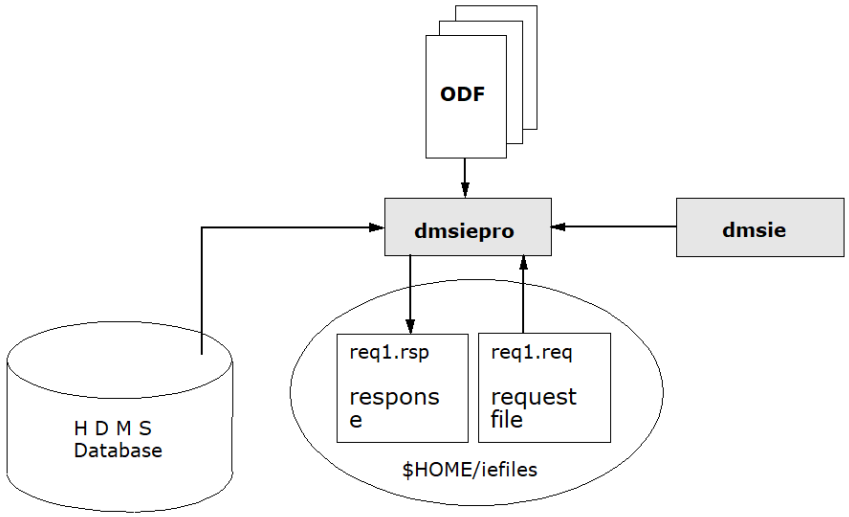

Figure 8: XIE-API Data Exchange via File Interfaces

The partner application generates a request file containing data exchange requests. The partner application then activates the mechanism for processing the requests in the request file. A distinction is made between two variants:

- Interactive call for `dmsieproc`: `dmsieproc` checks the access rights and executes the database requests. The results are stored in a response file. After dmsieproc ends, the partner application can read the request results from the response file.
- Copying of request files to `$HOME/iefiles`: The `dmsied` daemon process automatically forwards the request files in this directory to `dmsieproc`, where they are processed. The appropriate response files are then stored in the `$HOME/iefiles` directory. With this process, data from applications on remote computers can also be exchanged with OpenScape 4000 Manager/ Assistant.

6.2 Interface Description

The interface consists of the above-mentioned components, namely ODFs, request and response files and the `dmsieproc` and `dmsied` programs. These individual components are described below.

6.2.1 Object Description Files (ODF)

Please refer to Section 3.6, "Object Description Files (ODFs)"

6.2.2 Request and Response Files

ASCII files form the interface for exchanging data. The request and response files form this interface. The request file contains the requests sent to the OpenScape 4000 Manager/Assistant, while the response file contains the results of processing the requests. The client (partner application) assigns the request file name, which must contain the extension `.req`. The `dmsieproc` program creates the name of the corresponding response file based on the request file name by replacing the extension ".req" with ".rsp". The base name of both files is the same.

A request file can contain different import/export requests:

- Import requests

- Insert:	Inserts the specified data record
- Update:	Modifies an existing data record
- Delete:	Deletes an existing data record

- Export requests

- `Select:`	Reads a data record
- `Select_updates:`	Returns modified data records and the last operation performed on these records (`insert`, `delete` or `update`)

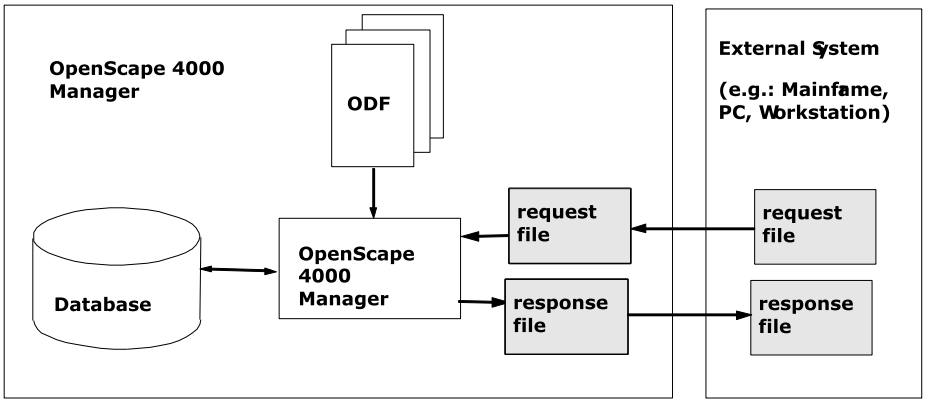

Figure 9: Request and Response Files

Request file

A request file consists of three parts, the header, the requests and the termination:

- Header The header consists of a comment character (#) and a string used for identification purposes. The header is the first line in a request file and must always be included, e.g.:

#ImportPerson;

- Requests The request consists of an integer identifier and the command part:
- `1;<command part>`

  The value must be between 1 and 2147483647. The identifier should be unique within a request file.

  You can use the `INSERT`, `DELETE`, `UPDATE`, `SELECT` and `SELECT_UPDATES` commands to formulate the command part:

  – `INSERT;odf_name;value1|....|valuen|` Inserts a transferred record into OpenScape 4000 Manager. All fields must always be specified. The first value must, if specified, be a unique key word.
  – `DELETE;odf_name;value1|....|valuen|` Deletes a transferred record from OpenScape 4000 Manager. You need only specify the value1 field, i.e. the unique key to identify the object to be deleted.
  – `UPDATE;odf_name;new_value1|...|new_valuen|` Replaces the value in the old record with the value in the new record. In order to use this operation, it must not be possible for an update request to change the values of the key attributes. The key attributes depend on the ODF. A unique internal key attribute (referred to as the `unique key`) is maintained for each object in the OpenScape 4000 Manager database. This internal value is specified as the first value in the standard ODFs in Chapter 10, "Description of ODFs". You must always specify all fields for the update command. If no value is specified, the associated attribute remains as it was. The first value must be a unique key word.

    NOTICE: To delete the text field, use a space character, e.g. `1;UPDATE;PERSDAT;53002|TR| |org1| org2|||||....`

  – `SELECT field1, field2,... FROM odf_name WHERE condition` Supplies all records which meet the specified condition. Only one condition can be specified here. The condition syntax is the same as the SQL syntax with one restriction: sub-queries are not allowed.
  – `SELECT_UPDATES field1, field2,... FROM odf_name WHERE LAST_UPDATE > time` Supplies all records which meet the specified condition (no further conditions may be specified!). You can replace the operator > in the WHERE clause with any other compare operator, The time of the last record changed is stored in `LAST_UPDATE`. You must specify `time` in the Informix date format `DATETIME to FRACTION(2)` (e.g. "2000-12-31 19:30:20.00"). "AND" links in the "WHERE" clause with the LAST_UPDATE attribute can only be entered. To obtain the modified data records, the flags for `port_update`, `pin_update` and

pers_update must be set to 001 in the admin table (refer to Section 3.7 on page 3-32).

Any number of requests can be formulated in the request part.

- Order identifier
- – #ORDER_ID=string The string may contain up to 6 characters. The order_id attribute from the ACTION_CTRL object description file is used for querying the order ID. The order_id attribute is assigned a value which is composed of the prefix string and the whole number identifier before the command part. If necessary, the order_id attribute can be increased to 12 digits by inserting leading zeros in front of the identifier. The identifier may not exceed 12 digits.
- Termination The termination consists of the string #@.

Response file

Like the request file, the response file also consists of three parts:

- Header The header consists of a comment character (#), the identifier transferred in the request file, the date and the time that processing of the corresponding request file began. If no identifier was specified in the request file, it is not included in the response file either. The header is the first line in a response file and is always included, e.g.:

```
#Import Person; 2000-12-31 20:15:12.11
```

1) The header can also consist of an ORDER_ID statement. This is then transferred back to the response file (as the header).

- Results The response file includes a result for each request in the request file. The result consists of the symbol #, the identifier (same as identifier in the request file) and the result of processing the request. The result composition depends on the command in the request file:
- If Informix errors occur, the relevant Informix error number is supplied in the response file (see Informix error messages). XIE-API errors are also supplied on a separate line in the response file, including the error number and corresponding error message.

  Return value if an error occurs: -52 - Error reading request file -53 - Incorrect operation -54 - Incorrect ODF name or lack of user rights for this ODF -55 - Incorrect ODF field name -56 - Operation not allowed for the specified ODF -57 - Syntax error -58 - Lack of resources -65 - DMS error message (followed by additional error numbers/error texts) from Configuration Management

  – INSERT: A negative number is supplied if an error occurs. If no error occurs, the number of inserted data records (1) is supplied. In addition, the value of the fields in the data records are supplied on a new line, separated by "|".

Positive:

```
#<id>;1 value1|value2|...|
valuen|
```

Negative:

```
#<id>;<neg.number>
```

– `DELETE, UPDATE:` If an error occurs, a negative number or zero is supplied. If no error occurs, the number of data records affected by the operation is supplied.

Positive:

```
#<id>;1
```

Negative:

```
#<id>;<neg.number or 0>
```

– `SELECT:` If an error occurs, a negative number or zero is supplied. If no error occurs, the number of supplied data records is returned. The records are supplied individually on the following lines, and the individual values in a record are separated by "|".

Positive:

```
#<id>;<m> value1|value2|...|valuen| ...
value<m>1|value<m>2|...|value<m>n|
```

Negative:

```
#<id>;<neg.number or 0>
```

– `SELECT_UPDATE:` If an error occurs, a negative number or zero is supplied. If no error occurs, the number (m) of supplied data records is returned. The records, along with the operation and the ODF, are supplied on the following lines. The syntax of this command is the same as the syntax in the request files. Possible commands are `insert`, `update` and `delete`. The `identifier` field is returned even if this field was left blank in the request file.

Positive:

```
#<id>;<m> INSERT;odf_name;unique_key|
value 1|value 2|...|value n| . . .
DELETE;odf_name;unique_key| . . .
UPDATE;odf_name;unique_key|value 1|value 2|...|
value n| . . .
```

Negative:

```
#<id>;<neg.number or 0>
```

- Termination The termination is on a separate line and consists of the string #@. If the termination is missing, this means that processing of the request file was terminated unexpectedly.

6.3 Data Exchange Examples

6.3.1 Personal Identification Number (PIN)

An application for calculating call charges needs information about all PINs in the Hicom network. You can read the data with a request file; the predefined ODF PIN is used (see Table 141 on page 284):
Request file (readpin.req)

```
#readPin;

1;SELECT * FROM PIN

#@
```

Process the request file by calling `dmsieproc readpin.req` or by copying the request file to `$HOME/iefiles` (`$HOME` is the home directory of the user who initiated the data export).
Response file (readpin.rsp):

```
#readPin; 2000-12-31 19:40:30.00

#1;30

122|MCH|31111|ANL1|123456789012|041|1|001|6|

      ... (30 records total)

#@
```

In the select command, you can also specify a simple WHERE clause which consists of the ODF attributes, compare operators and logical commands. You can replace the character "*" with a selection of ODF attributes, e.g.:
Request file (readpin.req):

```
#readPin;

1;SELECT switch_name, extension, pin_num from PIN
 WHERE
switch_name =

    "ANL1" AND pin_num > "100000000"

2;SELECT switch_name, extension FROM PIN where
 switch_name =
"ANL3"

#@
```

You can use the SELECT_UPDATES operation to locate all data records modified for this ODF, starting at a particular time:
Request File (selupdpin.req):

```
    #selupdPin;

    1;SELECT_UPDATES
 unique_key,switch_name,extension,pin_num
FROM

          PIN WHERE LAST_UPDATE > "2000-01-01 00:00:00.00"

    #@
```

Process the request file by calling `dmsieproc selupdpin.req` or by copying the request file to `$HOME/iefiles`.

When processing is complete, the response file shows the data records that have been both changed and deleted since the specified time.
`Response file (selupdpin.rsp):`

```
    #selupdPin; 2000-12-31 19:40:30.00

    #1;4

    INSERT;PIN;122|ANL1|31111|1234567890|

    INSERT;PIN;234|ANL1|11122|2222231212|

    DELETE;PIN;125|

    DELETE;PIN;129|

    #@
```

To obtain the deleted data records, the flags for `port_update`, `pin_update` and `pers_update` must be set to `001` in the `admin table`.

6.3.2 Personal Data

Certain data (particularly data relating to companies, organizations/ departments, locations, buildings, persons) can be imported into the OpenScape 4000 Manager/Assistant database from a remote application. The data to be imported is subjected to OpenScape 4000 Manager/Assistant Configuration Management tests which are also run on data entered via the interface.

Example: ODFs for personal data

Data relating to a company"s employee (name, title, relevant organization/ department, etc.) is stored on a host computer. This data is imported to the OpenScape 4000 Manager database to prevent it from being processed twice and also to ensure that it can be used in OpenScape 4000 Manager. The request file when importing data into the OpenScape 4000 Manager database for the first time is structured as follows for processing via XIE-API:
`Request file (imppers.req):`

```
# imppers
1;INSERT;PERSDAT;|AT|xyzcompany|EZE||||||Holzastner|
HOLZASTNER|H
annes|||Vie|||42534||
0|||||||||||||||||||||||||||||||||||||||
|||||||||||||||||||||||||||||||||
#@
```

If only specific attributes are to be read in from the ODF PERSDAT or if the sequence of attributes is to be changed, it is advisable to use a "derived ODF" which only contains the relevant attributes.

The data exchange is initiated by calling `dmsieproc imppers.req` or copying the file to `$HOME/iefiles`.

Once the data has been imported, the user receives a response file with the processing results.
Response file (imppers.rsp):

```
# Datenimport Person; 2000-10-29 19:40:30.00

#1;1

544|AT|xyzcompany|EZE||||||Holzastner|HOLZASTNER|Hannes|||
Vie|||
42534||0||||||||
20010719|||||||||||||||||||||||||||||||||||||
|||||||||||||||||||||||||||
#@
```

Personal data can be further processed via the ODF PERSDAT using the following operations: SELECT, SELECT_UPDATES, UPDATE, DELETE.

The internal unique_key assigned by OpenScape 4000 Manager when importing or creating a data record is used to identify a data record. Another key attribute can be selected with the aid of the derived ODF - XIE-API uses the attribute in the first position of the ODF as the unique_key.

6.3.3 Changing Display Names

The predefined ODF DISPLAY_NAME can be used to change a subscriber"s display name (MUSTERMANN* in the following example). In the request file, the Status field is taken from the list of coded values. The ORDER_ID can be used to verify processing at the OpenScape 4000 switch_name.
Request file (upddisplay.req):

```
#upddisplay;

#ORDER_ID=huber
```

```
1;UPDATE;DISPLAY_NAME;|ANL1|DOMAIN1|2000|MUSTERMANN*|ORG1|
1||
#@
```

Process the request file by calling `dmsieproc upddisplay.req` or by copying the request file to `$HOME/iefiles` (`$HOME` is the home directory of the user who initiates the data export).

Response file (upd`display.rsp`):

```
    #upddisplay; 2000-12-31 19:40:30.00

    #1;1

    #@
Request file for action control (select_oderid.req):



    #Get the OrderId

    #1; SELECT oder_id, operation, object_class,
 action_status,

        action_error FROM ACTION_CONTROL WHERE order_id
 matches
"huber*1234"

        and action_category=3

    #@
Response file for action control (select_oderid.rsp):



    #Get the OrderId; 2000-12-02 16:18:32.00

    #1;1

    huber0001234|649|601|067|065|

    #@
```

6.3.4 Verifying Integration at OpenScape 4000 System

The ODF for action control can be used for querying the status and the result of the order at the OpenScape 4000 system. The following action should be taken for this purpose.

- Execute request file with insert or update operation (with ORDER_ID)
- Perform periodic query with select request file in accordance with corresponding ORDER_ID to action_status="067"
- Verify the `action_error` field.

The following short shell program example illustrates the principle of the procedure.

Example: Script file for verifying integration at OpenScape 4000 system

```
#!/bin/csh

/opt/xie/dmsie/bin/dmsieproc upddisplay.req

set Result=$status

if ($Result != 0) then

   echo Error: dmsieproc - return-code: $Result

   exit $Result

endif

set Action_Status="064"

while ( $Action_Status != "067" )

  echo "Job running ..."

  /opt/xie/dmsie/bin/dmsieproc select_orderid.req

  set Action_Status=`awk -F\| ' ! /#/ {printf "%s\n", $4 } '
select_orderid.rsp`

  sleep 1

end

echo "Job completed ..."

set Action_Error=`awk -F\| ' ! /#/ {printf "%s\n", $5 } '
select_orderid.rsp`

switch ( $Action_Error)

  case "065":

       echo "OK"

       breaksw

  case "063":

       echo "DMS-Error"

       breaksw

  case "066":

       echo "Error"
```

```
      breaksw
  default:
      echo action_error=$Action_Error
endsw
```

6.3.5 Derived ODFs

`Derived ODFs` are based on the standard ODFs described in Section 3.6, "Object Description Files (ODFs)" and Chapter 10, "Description of ODFs". The number of derived ODF fields is a subset of the standard ODF fields. The number of customer-specific ODF fields is a subset of the standard ODF fields. All field names specified in the derived ODFs must therefore also be included in the corresponding standard ODFs.

Derived ODFs are contained in the delivery of the XIE-API package, but can also be generated by a system administrator. By creating Derived ODFs (customer-specific ODFs), the user is able to create object descriptions which are tailored to his special needs.

The system administrator can generate derived ODFs in the `/opt/xie/dmsie/odf/drv` directory and has to create corresponding mapping files in the `/opt/xie/dmsie/odf/mpp` directory using any text editor. The user of the derived ODFs must have got assigned the feature of the original ODF to be able to use the derived ODF.

Syntax description

In the syntax description below, key words are printed in `boldface` and optional parameters are enclosed in `[` and `]`. Alternatives are separated by `|`.

```
<derived_ODF>::=    [<export_operations><newline>]
                    [<import_operations><newline>]
                    <field_defs><field_end> <std_ODF_name><end>

<export_operations>::=<delimiter>EXPORT<export_op>

<import_operations>::=<delimiter>IMPORT<import_op>

<field_defs>::=<field_def>[<field_defs>]

<field_def>::=<field_name><spaces><field_type><newline>

<spaces>            one or more blanks

<delimiter>::=:

<export_op>::=[:SELECT] [:SELECT_UPDATES]

<import_op>::=[:INSERT] [:DELETE] [:UPDATE]

<field_name>        string containing the field name (see Chapter 10, "Description of ODFs")
```

```
<field_type>::=     CHAR[ACTER](<posint>) |
                    INT |
<posint>::=integer between 1 and 32511
<field_end>::=&&<newline>
<std_ODF_name>      standard ODF name from which this ODF was derived
<end>::=@@
```

Example: Derived ODF

A user needs the `company`, `country`, `name` and `switch_name` data fields from the `PERSDAT` table. However, the user should not be able to access any of the other data fields in ODF `PERSDAT` for reasons of data security. The system administrator therefore generates the `DRV_PERS` file with the following contents in the `$XIEHOME/dmsie/odf/drv` directory:

```
# DRV_PERS        char(30)
:EXPORT:SELECT    char(2)
company           char(25)
country           char(4)
name
switch_name
&&
PERSDAT
@@
```

The desired data can be read with the following request file:
`Request file (getpers.req):`

```
#all Pers;
1;SELECT * FROM DRV_PERS
#@
```

Process the request file by calling `dmsieproc getpers.req` or by copying the request file to `$HOME/iefiles`.
`Response file (getpers.rsp):`

```
#all Pers; 19.10.2000-18:20:30.00
#1;1500
company1|country1|name1|SWI1|
```

```
  . . . (1500 records total)

#@
```

7 API Import/Export Client

The API Import/Export Client permits the exchange of data between the OpenScape 4000 databases and partner applications.

7.1 General Information

The API Import/Export Client is a Windows application with the following features:

- Display a login window; the user logs on to the OpenScape 4000 Manager/ Assistant with username and password
- User Interface
- Export data to screen or hard disk
- Import data from hard disk
- Schedule export or import jobs

The Import/Export Client can be started via Start Desktop Main Window ("Import/Export Table" item in the application tree), assuming that the user has the appropriate access rights for this feature.

Upon starting the Import/Export Client, the authentication box is displayed first, whereby the user must log on to the OpenScape 4000 Manager/Assistant with username and password (see Figure 10).

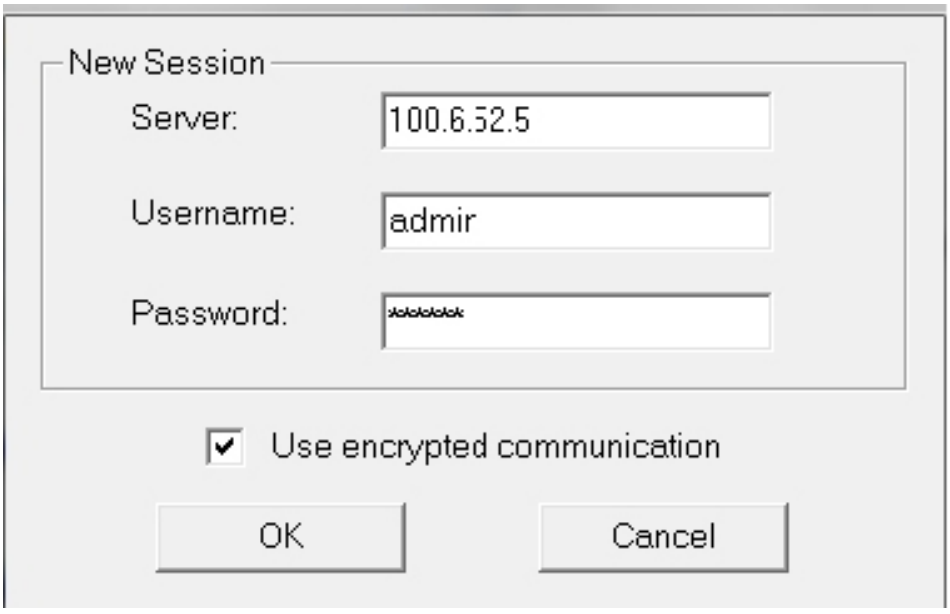

Figure 10: Authentication Box

The checkbox is marked, all data between client and server are transferred via TLSv1.

After user verification, the Import/Export Client window is displayed. This window is divided into the following parts (see Figure 11):

- Menu bar - Contains the required commands
- Tool bar - Provides a shortcut for most functions used
- Property sheet - For switching between the following tab pages:

- – Export Table page (refer to Section 7.2)
  – Import Table page (refer to Section 7.3)
  – Scheduler page (refer to Section 7.4)
- Status bar - Displays the current status

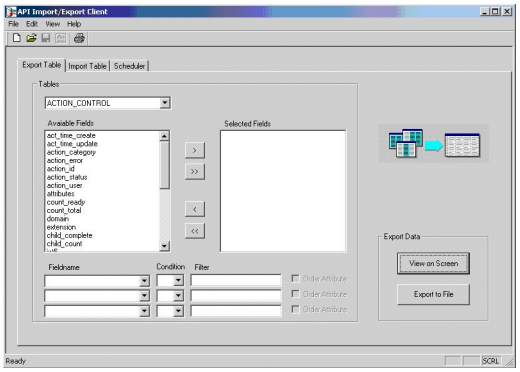

Figure 11: Main Screen of OpenScape 4000 Manager Import/Export Client

7.2 Export Table Page

The Export Table page offers features for:

- Selecting a table from the OpenScape 4000 database
- Choosing fields from the selected table
- Defining filter criteria and order attributes
- Viewing the desired table records on the screen
- Storing records on the PC's hard disk
- Printing records on a standard Windows printer

7.2.1 The File Menu

The File menu for the Export Table page is shown in Figure 12.

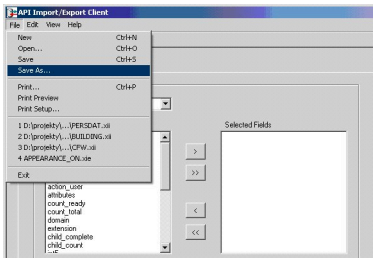

Figure 12: File Menu

Choosing File from the Export Table main menu bar allows you to access windows to do the following:

- New - Resets everything in the currently selected property page.
- Open - Opens a defined selection that was previously saved.
- Save - Displays the standard Windows Save dialog box to save defined selections; for the export table, saved files have the extension .xie.

- Save As - Displays the standard Windows Save As... dialog box to save defined selections; for the export table, saved files have the extension .xie (see Figure 12).
- Print - Prints the records.
- Print Preview - Displays the standard Windows Print Preview dialog box.
- Print Setup - Displays the standard Windows Print Setup... dialog box.
- n *filename (example:* 1 PERSDAT.xie) - Reopens a previously defined selection.
- Exit - Exits the API Import/Export Client application.

7.2.2 The Edit Menu

The Edit menu for the Export Table page is shown in Figure 13.

Figure 13: Edit Menu

Choosing Edit from the Export Table main menu bar allows you to do the following:

- View on screen - Display records on the screen.
- Export to File - Exportsrecords to a file.

7.2.3 The View Menu

The View menu for the Export Table page is shown in Figure 14.

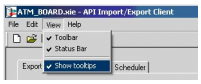

Figure 14: View Menu

Choosing View from the Export Table main menu bar allows you to do the following:

- Toolbar - Turns the toolbar on or off.
- Status Bar - Turns the status bar on or off.
- Show tooltips - Turns the tooltip on or off.

7.2.4 The Help Menu

The Help menu for the Export Table page is shown in Figure 15.

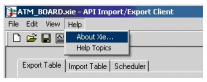

Figure 15: Help Menu

Choosing Help from the Export Table main menu bar allows you to do the following:

- About Xie ... - Displays the About Export Table dialog box.

- Help topics - Opens online Help

7.2.5 The Tool Bar

Figure 16: Toolbar

The most frequently used functions can be accessed directly via the buttons on the Export Table window toolbar:

New - This button resets everything in the window

Open - This button opens a standard Windows open dialog for a saved query

Save - This button saves the records on the PC's hard disk

Print - This button prints the records on Windows Printers

View - This button shows the records on the screen

7.2.6 Export Table Property Page Window

The Export Table property page window (see Figure 17) allows you to request data from the OpenScape 4000 Manager common database.

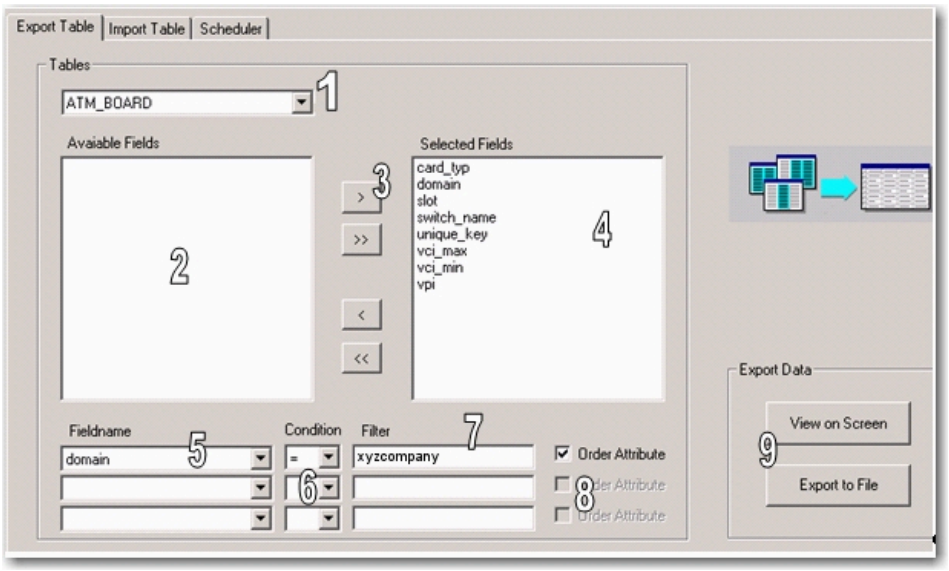

Figure 17: Export Table Property Page

The combo box (1) is used to select a OpenScape 4000 Manager table from the OpenScape 4000 Manager common database. Depending on the user that is logged in, all valid OpenScape 4000 Manager tables are listed in alphabetical order in the combo box. The default value is the first OpenScape 4000 Manager table in the list.

After a table is selected, the available fields are shown in the left list box (2). To select a field, double-click an available field or select the field using the buttons between the two list boxes (3). The fields you have chosen are shown in the Selected Fields list box (4).

Filter conditions are used to limit the number of records affected by an operation, and to sort output. To filter output, select a field name from the Fieldname combo box (5), a condition (6) (equality is the default condition; also available are inequality, greater than or equal to, or less than or equal to), and the appropriate filter value.

The filter (7) value can contain the wildcard characters * and ?. Handling wildcard characters for database access is the same as for executing structured query language (SQL) commands.

In SQL, you cannot combine "greater than or equal to" or "less than or equal to" with a wildcard character. If you try to do this, the program changes the condition to equality.

Because of the restriction that null values are not listed when asking for all values using a *, the null values can be selected using the condition != and the filter *.

If you specify more than one Fieldname/Filter Condition (a maximum of three), the individual fields are linked by the AND operator.

If you select the Order Attribute check box (8), the result records are sorted in ascending order.

You can also sort records based on a field name that is not limited by conditions. In this case, you can enter the field name in the Fieldname combo box, but leave the Condition and Filter combo boxes blank. You must select the Order Attribute check box.

When you click the View on Screen button (9), the dialog box shown in Figure 18 is displayed.

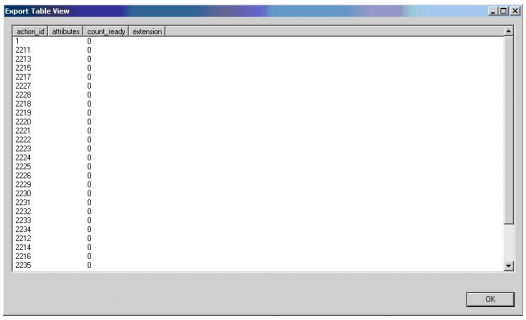

Figure 18: Example of OpenScape 4000 Manager Export Table View

To close the OpenScape 4000 Manager Export Table View dialog box, click OK.

Click the Export to File button (9) to write records to a file stored on the OpenScape 4000 Manager Desktop PC hard disk (see Figure 19).

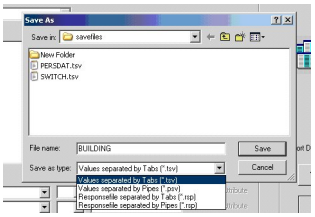

Figure 19: Exporting Data to a File

Exporting data to a file on hard disk is available in the following formats:

- Save as a response file, values separated by tabs
- Save as a response file, values separated by pipes (|)
- Save as a text file, values separated by tabs
- Save as a text file, values separated by pipes (|)

The following are examples of each type of export format.

Example: Response file with values separated by tabs (*.rsp)

```
#DOMAIN; created on Wednesday, February 10, 2000 at
 15:01:14

#1;3

101     ERD  info about domain ERD

208     SIE  info about domain SIE

507     GUD  info about domain GUD

#@
```

Example: Response file with values separated by pipes (*.rsp)

```
#DOMAIN; created on Wednesday, February 10, 2000 at
 15:01:14
```

```
#1;3

101|ERD|info about domain ERD|

208|SIE|info about domain SIE|

507|GUD|info about domain GUD|

#@
```

Example: Text file with values separated by tabs (*.tsv)

```
101     ERD  info about domain ERD

208     SIE  info about domain SIE

507     GUD  info about domain GUD
```

Example: Text file with values separated by pipes (*.psv)

```
101|ERD|info about domain ERD|

208|SIE|info about domain SIE|

507|GUD|info about domain GUD|
```

7.3 Import Table Page

The Import Function of the XIE Client allows data to be imported from a text file. When the Client starts up, the user must enter the OpenScape4000 Server name and the user/password combination. After authentication, the main screen is displayed by clicking the Import Table tab. If you change something in the current page, you will be asked by a message box to save the changes (see Figure 20); otherwise the changes will be ignored.

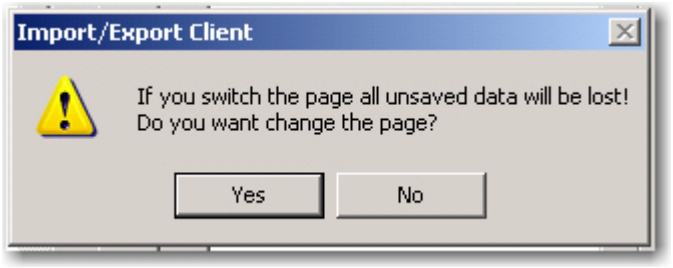

Figure 20: Switching Tab Pages in the Import/Export Client

The Import Table page offers following features:

- Naming of the table to which data will be imported
- Inserting, updating, and deleting of records
- Naming of the file that is to be imported

- Specifying the delimiter used between data fields (the sentences should be line oriented)
- Selecting "First line includes field names"
- For each data field: name of the field within the defined HDMS table to which it should be assigned
- Selecting "Import "

7.3.1 The File Menu

The File menu for the Import Table page is the same as for the Export Table page (see Figure 12), except that the save files for Save and Save As ... functions have the extension .xii instead of .xie.

7.3.2 The Edit Menu

The Edit menu for the Export Table page is shown in Figure 13.

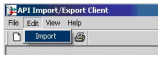

Figure 21: Edit Menu

Choosing Edit from the Import Table main menu bar allows you to do the following:

- Import - Import defined query into database

7.3.3 The View Menu

The View menu for the Import Table page is the same as for the Export Table page (see Figure 14).

7.3.4 The Help Menu

The Help menu for the Import Table page is the same as for the Export Table page (see Figure 15).

7.3.5 The Tool Bar

Figure 22: Toolbar

The most frequently used functions can be accessed directly via the buttons on the Import Table window toolbar:

New - This button resets everything in the window

Open - This button opens a standard Windows open dialog for a saved query

Save - This button is disabled for the Import Table assistant.

Print - This button is disabled for the Import Table assistant.

View - This buttonis disabled for the Import Table assistant.

7.3.6 Import Table Property Page Window

The Import Table property page window (see Figure 23) allows you to import data to the OpenScape 4000 Manager common database.

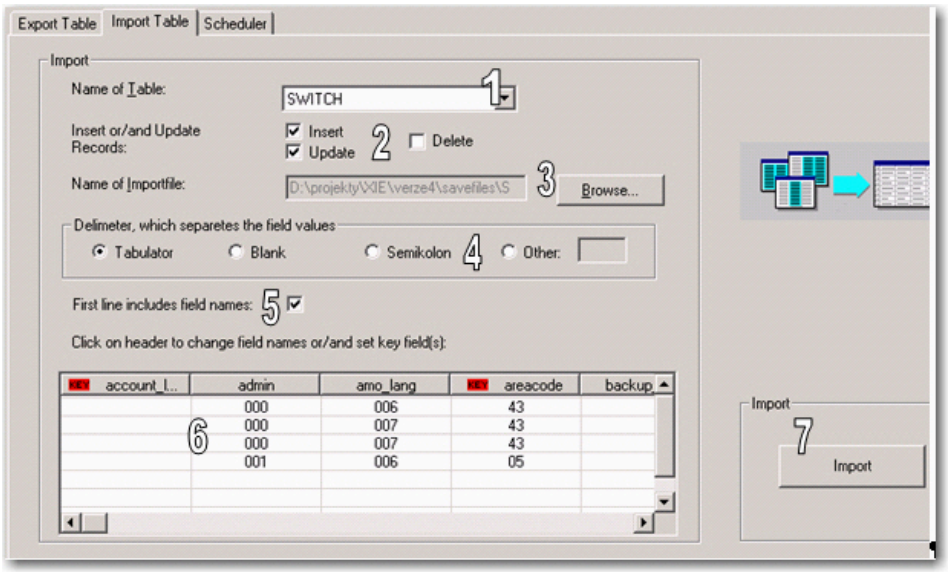

Figure 23: Import Table Property Page

The Combo box (1) is used to select the Table/Object file into which the data will be imported. The following Table/Object files are available for import:

PERSDAT, BUILDING, FIRMDAT, ORG, LOCDAT, CTRYNUM, CFW, COS, DISPLAY_NAME, DOMAIN, PEN, PIN, PORT, PRIV_DIAL_NUMBER, SWITCH, KZNR, KZNR_LIST

Next, the type of database operation is selected (2):

- Update - the old data are overwritten.
- Insert - the new data are inserted.
- Delete - the selected data are deleted.
- Update and Insert - the new data are inserted if they are not in the database.

NOTICE: If the data are already in the database, they are updated. In case of an update, values that are not contained in the import file remain unchanged. If more than 100 records are updated or deleted, a warning is displayed and you can cancel or continue the modifications.

The Browse button (3) next to the input field for the name of the import file enables you to find and select a file in the file system. This file must be in an ASCII format (see Figure 24).

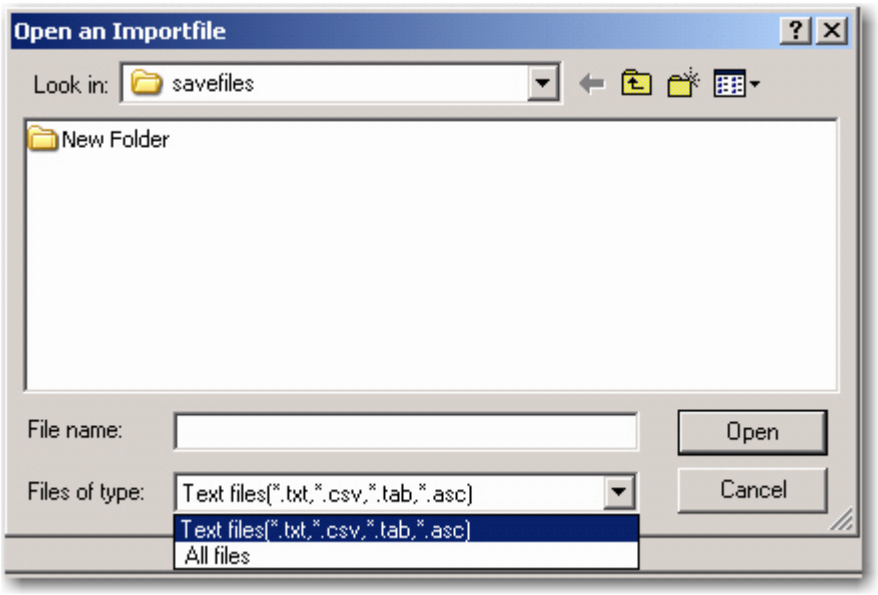

Figure 24: Selecting an Import File

Then you must select a separator (4) between entries in the ASCII file. The separator can be one of the predefined values (tabulator, blank, semicolon) or a different character, which you must write into the Other field. If the first line of your ASCII file is used to specify the field names, you should select checkbox (5).

Example: ASCII file using pipe as separator and first data line contains field names from PERSDAT table

```
company|country|name|location|
xyzcompany|D|Vogell|MUE H|
xyzcompany|D|Aberer|MCH H||
xyzcompany|H|Barbarics|SYSDATA|
xyzcompany|A|Lukas|VIE MOS|
||Vogell||
||Aberer||
```

The content of the file is previewed in the list box (6). The list box shows the first 15 data sentences of the import file.

If the first line of the import file contains the field names (option First line includes field names is selected), these are displayed as the corresponding column headings in the first line of the list box. If a given field name does not

correspond to a field name of the selected OpenScape4000 Table, the field name is written in red and must be changed before import.

If the first line of the import file does not contain the field names, or if some of the field names are not correct (printed in red), the fields for the selected table can be displayed in a pop-up menu (by clicking the right mouse button) and corrected by selecting the field names accordingly (see Figure 25).

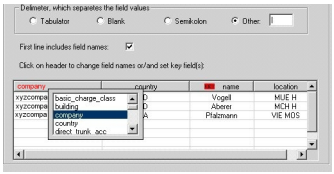

Figure 25: Editing Field Names in the Import Table

As the number of the fields to be imported may exceed the available space in the window, a scroll function makes it possible to handle all fields for the selected table at once.

If you need run an update operation on the database, you must specify the "key" (or optionally multiple keys). This can be done by left-clicking the field name in the column heading.

If you want to run a delete, every field is marked as a key value. For security reasons, you cannot switch between "key" and "no key" in the case of a delete operation.

Upon completing the steps as described above, you can import the data by clicking the Import button (7).

During the import operation, a report is generated which indicates the success or failure of each import or update operation. The report is stored in a file named <inputfilename>.log in the same directory where the ASCII file for import is located. The user is responsible for deleting report files.

After import, the user sees a message box with a summary of errors that occurred, if any (see Figure 26).

Figure 26: Message Box on Completion of Import

Clicking Yes in this message box displays the complete report file (see Figure 27).

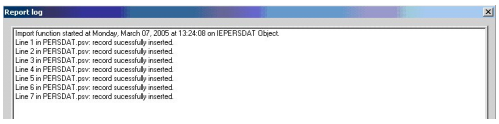

Figure 27: Complete Report Log

7.4 Scheduler Page

The Scheduler function of the XIE Client allows you to schedule Export or Import jobs. After authentication, the Scheduler main screen is displayed by clicking the Scheduler tab (see Figure 30).

The Scheduler page offers the following features:

- Name of the file which includes import or export query
- Destination directory for saving the results of jobs
- Starting date and time, ending date (optional)
- Frequency of executing saved task
- The task name
- Should be the task name replaced if it exists
- Select "Schedule"

7.4.1 The File Menu

The File menu for the Scheduler page is the same as for the Export Table page (see Figure 12), except that the save files for Save and Save As ... functions have the extension .xis.

7.4.2 The Edit Menu

The Edit menu for the Scheduler page is shown in Figure 28.

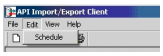

Figure 28: Edit Menu

Choosing Edit from the Import Table main menu bar allows you to do the following:

- Schedule - Schedule defined job

7.4.3 The View Menu

The View menu for the Scheduler page is the same as for the Export Table page (see Figure 14).

7.4.4 The Help Menu

The Help menu for the Scheduler page is the same as for the Export Table page (see Figure 15).

7.4.5 The Tool Bar

Figure 29: Toolbar

The most frequently used functions can be accessed directly via the buttons on the Scheduler window toolbar:

New - This button resets everything in the window

Open - This button opens a standard Windows open dialog for a saved query

Save - This button is disabled for the Scheduler.

Print - This button is disabled for the Scheduler.

View - This button is disabled for the Scheduler.

7.4.6 Scheduler Property Page Window

The Scheduler property page window (see Figure 30) allows you to schedule Import and Export jobs.

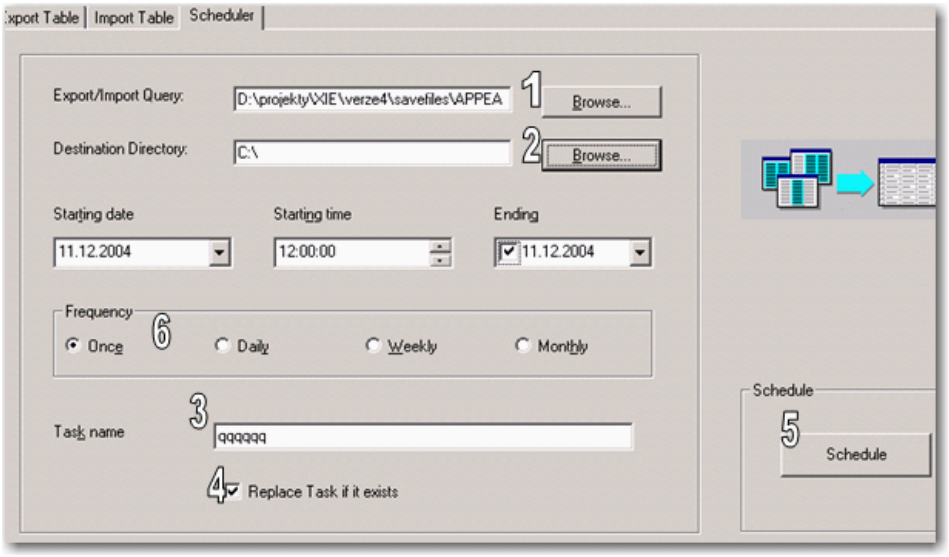

Figure 30: Scheduler Property Page

First, click the Browse button (1) to select the Export Table query or the Import Instruction that is to be processed.

In case of an Export Table query, the destination directory for the exported file must be specified. The results of the query will be saved in this directory. You can also select this directory using the corresponding Browse button (2).

Next, select the Starting date and Starting time (the date and time a one-time export job will run, or the beginning date and time for a repeating job). The

starting and ending dates are entered via Date Selection Control (see Figure 31):

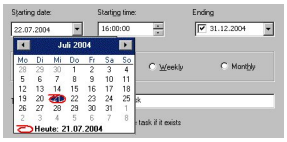

Figure 31: Specifying the Start and End of Import/Export Jobs

> **NOTICE:** The ending date is the last day that a repeating job will run (optional, N/A for one-time jobs). If this is not set for a repeating job, the job will repeat indefinitely.

Then, select the Frequency (6) with which the job will be executed (One time, Daily, Weekly or Monthly).

The Task name (3) is a string which will be displayed in the task's property sheet in the scheduler UI. Figure 32 shows the added job (named "Sample Import Task") in the Microsoft (Windows 2000 or XP) scheduler UI:

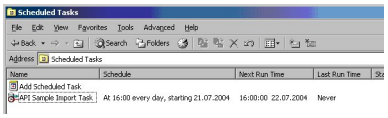

Figure 32: Task Scheduler

If a task with the same name already exists, you can overwrite it by selecting the check box (4).

You can then schedule this job by clicking the Schedule button (5).

The Microsoft scheduler API is then displayed (see Figure 33):

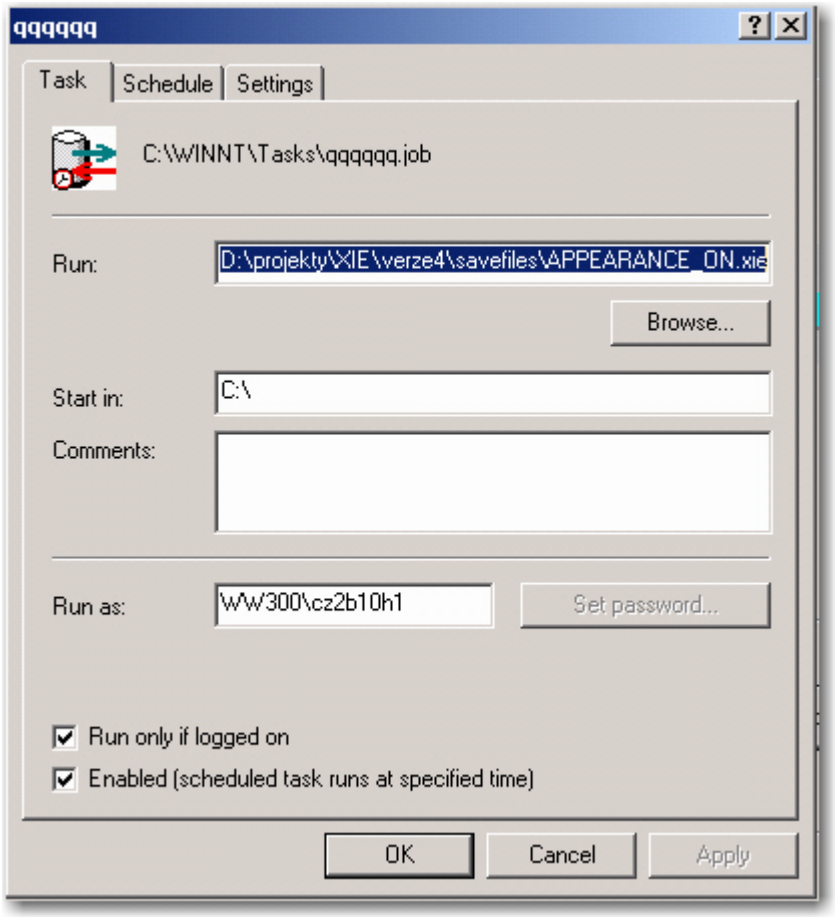

Figure 33: Scheduler API

With this API you can change or set more instuctions for running your job. You can also set your account and password for running a job if you are logged off (Run as and Set Password). The default setting for the API Import/Export Client is Run only if logged on (check box selected).

The Microsoft scheduler user interface (START/Settings/Control Panel/ Scheduled Tasks) can also be used for subsequent job manipulation (listing jobs, deleting jobs, changing date and/or time, changing frequency, e.g., to every 10 minutes or one hour).

Example: Repeat a task every 60 minutes (see Figure 34**)**

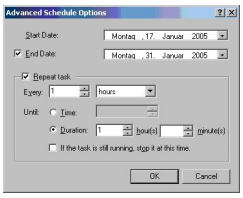

Figure 34: Advanced Schedule Options

8 The api2hipath Command Line Tool

The api2hipath command line tool allows you to export tables from the OpenScape 4000 common database to an ASCII file and import data from an ASCII file to the OpenScape 4000 common database.

The Windows-version api2hipath command line tool is installed during the Export Table installation; the Linux version is located in the /opt/xie/dmsie/bin directory on OpenScape 4000 Manager/Assistant.

The number and layout of these tables corresponds to the available ODFs. Refer to Chapter 10, "Description of ODFs" for available ODFs and assigned attributes.

8.1 Exporting Data to an ASCII File

Syntax:

```
api2hipath -s fields [-w clause] [-t] [-n] [-z] -l user -p passwd [-h serverip] -
o
odfname [-c sep] filename
```

Table 4: api2hipath Export Syntax

Element	Description
-s fields	A ODF field or fields that you want to export. Each field has to be separated by comma. An asterik (*) indicates all fields in the specified ODF will be exported.
-w clause	Where-clause in SQL syntax (e.g. "switch_name = "SW10" AND name matches "A*" "). If no where-clause is specified, every row of the ODF is exported
-t	Convert every character in the output-file to upper-case letters
-n	No header line with fieldnames is written to the output-file
-z	Use encrypted communication with server (TLSv1)
-l user	OpenScape 4000 username with API permission
-p passwd	OpenScape 4000 password
-h serverip	OpenScape 4000 server IP address, an asterisk (*) indicates the local OpenScape server; default is the local server
-o odfname	API ODF name, Refer to Chapter 10, "Description of ODFs" for available ODFs
-c sep	Character that separates the data fields in each line of the output file. The default separator is comma (,)
filename	This output file receives the selected rows from the ODF during the export operation. The default pathname for the output file is the current directory

Example:

The following statement exports the fields name, christianname and extension from the ODF PERSDAT where the value of switch_name is equal to KN12 and puts them in a file named persdat.txt:

```
api2hipath -s "name,christianname,extension" -w "switch_name = "KN12"" -l user -
p
usrpasswd -o PERSDAT -c "|" persdat.txt
```

The output file, persdat.txt, appears as shown in the following example:

```
name|christianname|extension| Holzastner|Hannes|42534|
Prikop|Martina|42558| ...
```

Formatted output:

The data can be formatted by adding a "format" clause to each field in the -s option:

fieldname[=format]

The format consists of zero or more ordinary characters (not %) which are directly copied to the output, and zero or one conversion specification, which is introduced by a % and completed by the character s.

The conversion specification takes the following general form and sequence:

%[-][0][width][.precision]s

Element	Description
%	Always located at the beginning of the conversion specification. If the % character is not to be interpreted as a part of the conversion specification but as an ordinary character to be output, it must be escaped by preceding it with another % (%%).
-	The result of the conversion will be left-justified within the field. (It will be right-justified if this flag is not specified.)
0	Leading zeros are used to pad to the field width; no space padding is performed. If the 0 and - flags both appear, the 0 flag will be ignored
width	An optional entry that specifies the minimum field width: If the converted value has fewer characters than the field width, it will be padded (with space by default) on the left or right (see the above flag description) to the field width.
.precision	It specifies the maximum number of bytes to output. All characters from the field are output until the number of characters specified for precision is reached. The default is to take the entire string.
s	Always located at the end of the conversion specification

Examples:

Output field "name" in a 12-character field, left-adjusted:

```
api2hipath -s "name=%-12.12s" ...
```

Output the first 6 characters of field "name" with an appropriate message:

```
api2hipath -s 'name=the first 6 characters of name are
 %.6s' ...
```

Output a field "persnum" in an 15-character field right-adjusted witch leading zeros:

```
api2hipath -s "persnum=%015s" ...
```

8.2 Delta-Export Data to an ASCII File

Delta export writes all modifications on a ODF during a given time interval to an ASCII file.

Syntax:

```
api2hipath -s fields [-w clause] -f timefrom [-e timeto] [-t] [-n] [-z] -l user -
p passwd
[-h serverip] -o odfname [-c sep] filename
```

Table 5: api2hipath Delta Export Syntax

Element	Description
-s fields	A ODF field or fields that you want to export. Each field has to be separated by comma. An asterik (*) indicates all fields in the specified ODF will be exported.
-w clause	Where-clause in SQL syntax (e.g. "switch_name = "SW1" AND name matches "A*" "). If no where-clause is specified, every row of the ODF is exported
-f timefrom	Start date and time of delta export. Syntax: "YYYY-MM-DD hh:mm:ss"
-e timeto	End date and time of delta export. Syntax: "YYYY-MM-DD hh:mm:ss"; default is the current date and time
-t	Convert every character in the output-file to upper-case letters
-n	No header line with fieldnames is written to the output-file
-z	Use encrypted communication with server (TLSv1)
-l user	OpenScape 4000 username with API permission
-p passwd	OpenScape 4000 password
-h serverip	OpenScape 4000 IP address, an asterisk (*) indicates the local OpenScape server; default is the local server
-o odfname	API ODF name, only the ODFs PERSDAT, PORT and PIN are supported for delta-export
-c sep	Character that separates the data fields in each line of the output file. The default separator is comma (,)
filename	This output file receives the selected rows from the ODF during the delta export operation. The default pathname for the output file is the current directory. In the first column the type of operation is written (INSERT, UPDATE or DELETE, in case of DELETE in the second column the unique_key is written)

Example:

The following statement exports the fields name, christianname and extension from the ODF PERSDAT. All records modified/updated or deleted after 1 June 2012, 22:00 will be exported in a file named persdat.txt:

```
api2hipath -s "name,christianname,extension" -f "2012-06-01 22:00:00" -l user -p
usrpasswd -o PERSDAT -c "|" persdat.txt
```

The output file, persdat.txt, appears as shown in the following example:

```
OPERATION|name|christianname|extension| INSERT|Hracek|Petr|
42534| UPDATE|Jirout|Martina|42534| DELETE|2011|
...
```

8.3 Importing Data from an ASCII File

Syntax:

```
api2hipath -i / -ui / -u / -d [-w clause] -l user -p passwd [-z] [-h serverip] -o
 odfname
[-c sep] filename
```

Example 1:

Table 6: api2hipath Import Syntax

Element	Description
-i	Insert new record(s) from input file. The user has to have the API Access Right "API Insert operation allowed" assigned - refer to Section C.5
-ui	Update record(s) from input file, if the key-value does not exist the record will be inserted. The key-field has to be marked with an asterisk (*) in the input file. The user has to have the API Access Right "API Insert operation allowed" and "API Update operation allowed" assigned - refer to Section C.5
-u	Update record(s), the key-field has to be marked with an asterisk (*) in the input file. The user has to have the API Access Right "API Update operation allowed"assigned - refer to Section C.5
-d	Delete record, every field in the input file is regarded as key, except fields marked with a minus (-). The user has to have the API Access Right "API Delete operation allowed" assigned - refer to Section C.5
-uid	Update ,insert or delete records. The first column of the input file has to contain the fieldname OPERATION (in upper case letters) and the value of this field has to be the type of the operation (UPDATE, INSERT or DELETE). Refer to Example 2 following this table.
-w clause	optional update/delete where-clause in SQL syntax (e.g. "switch_name = "SW10" AND name matches "A*" ").
-l user	OpenScape 4000 username with API permission
-p passwd	OpenScape 4000 password
-z	Use encrypted communication with server (TLSv1)

Element	Description
-h serverip	OpenScape 4000 IP address, an asterisk (*) indicates the local OpenScape server; default is the local server
-o odfname	API ODF name, refer to Chapter 10, "Description of ODFs" for available ODFs with import possibility (e.g. PERSDAT, PORT, DISPLAY_NAME)
-r orderid	orderid, default is "api2hipath"
-c sep	Character that separates the data fields in each line of the output file. The default separator is comma (,). You cannot use the following items as delimiter characters: asterisk (*), minus (-) and new-line character
filename	This input file contains the data to be imported into the specified ODF. The default pathname for the input file is the current directory The first line has to contain the field names, separated by the delimiter character. Key fields have to be marked with a preceding asterisk (*), fields with preceding minus (-) will be ignored. The following lines have to contain the field values, separated by the delimiter character. New-line character and the delimiter character cannot be used in field-values, the character vertical bar (|) will be converted into slash (/). Assigning a NULL value to a field can be done by two delimiter characters with nothing between them.
-z	Use encrypted communication with server (TLSv1)

The following example shows the contents of a hypothetical input file named persdat.txt:

```
name|christianname|*extension|*switch_name|-persnummer|
Holzastner|Hannes|42534|KN12|0815| Prikop|Martina|42558|
KN12|0816|
```

This data file conveys the following information:

- 2 records with values for field name, christianname, extension, switch_name and persnumber
- The fields extension and switch_name are defined as key fields (marked with an preceded asterik in the first line of input file)
- The field persnumber should be ignored (marked with an minus in first line)
- Uses the vertical bar (|) for delimiter character

The following statement updates/inserts the fields name, christianname, extension and switch_name in the ODF PERSDAT:

```
api2hipath -ui -l usr -p usrpasswd -h serverip -o PERSDAT -c "|" persdat.txt
```

The following statement deletes records from the ODF PERSDAT - key values are name, christianname, extension and switchname (every field in the input file is regarded as key-field, except fields with preceding minus):

```
api2hipath -d -l usr -p usrpasswd -h serverip -o PERSDAT -c "|" persdat.txt
```

Example 2:

The following example shows the contents of a hypothetical input file named persdat2.txt:

`OPERATION|name|christianname|*extension| INSERT|Jirout| Petr|42534| UPDATE|Duda|Jan|42558| DELETE|Hracek|Petr| 50001|` This data file conveys the following information:

- 3 records with values for field name, christianname, extension
- The field extension is defined as key field
- The 1st record should be inserted, the 2nd updated and the 3rd deleted
- Uses the vertical bar (|) for delimiter character

The following statement updates/inserts/deletes records in the ODF PERSDAT:

```
api2hipath -uid -l usr -p usrpasswd -h serverip -o PERSDAT -c "|" persdat2.txt
```

8.4 Return values of api2hipath

Table 7: api2hipath Exit Codes

exit code	Description
0	OK
1	Wrong or missing parameter
2	Wrong user and/or password
3	No access to ODF or unknown or wrong ODF-name
4	Wrong syntax of time format in case of delta query
5	Cannot create output/input file (wrong filename or no permission)
6	Key-field not defined in case of update operation
7	Error on insert, refer to error text written on stderr
8	Error on update, refer to error text written on stderr
9	Error on delete, refer to error text written on stderr
11	No connection to OpenScape 4000 server (xieserver not running, wrong server IP address, max. number of clients exceeded), refer to error text written on stderr
12	User has no access rights on ODF
13	Operation insert or update or delete not allowed on ODF or no API license installed
14	Unknown/wrong fieldname
> 14	Other error, refer to error text written on stderr

9 XIE Web Service

The XIE Web Service interface provides encrypted communication with the XIE server via a set of web service operations.

Requests to XIE over the web service have a similar format as the requests in the XIE File Interface.

A web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the web web service in a manner as determined by its description using SOAP messages, typically conveyed using HTTPS with an XML serialization in conjunction with other web-related standards.

The communication via the XIE Web Service interface is suitable for modern external applications.

The XIE Web Service is an API, i.e. users of this feature are application developers and there is no graphical user interface.

The API is defined by the web service description file (WSDL). Users are able to download the WSDL from the OpenScape 4000 Manager/Assistant, section Utilities->XIE web service.

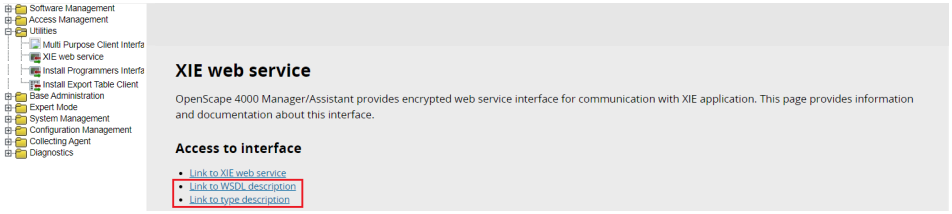

Figure 35: WDSL Description Download

9.1 Access to the interface

The name of the web service is XIEservice, and it is accessible via it"s URL:

```
https://<IP>/WS/XIEservice
```

The GUI of the OpenScape 4000 Assistant/Manager contains a link to the entry point of the XIE Web Service. The page is accessible from Utilities -> XIE web service.

The WSDL and type description are automatically generated during the start of the web service and can be downloaded from the OpenScape system using the following URLs:

```
https://<IP>/WS/XIEservice?wsdl

https://<IP>/WS/XIEservice?xsd=1
```

XIE Import/Export API and Configuration Management must be enabled in Application Control for the interface to function properly.

9.2 Security

The XIE interface is accessible via HTTPS; the access is protected by Security Management. Security Management is used to identify the client. A valid session cookie is required.

A user with a "cust" security profile has to have access rights to the "Import/Export API (XIE)" in "Access Right Configuration" application. This access right can be further specified to provide access only to specific ODF files in "Access Right Group Configuration". Users with the profiles "engr", "rsta", "rsca" and "cusa" have access to all ODF files and operations.

9.3 List of web service operations

The XIE Web Service provides the following operations:

Object Data Files info

- getODFNames() - get list of all ODF files which are accessible to the current user
- getODFContent() - get the content of a specific ODF file

Object Data Files request control

- processData() - create request to XIE server, result contains request ID for further processing
- getStatus() - get status of specific request defined by ID
- fetch() - fetch result data from beginning from specific request defined by ID
- fetchNext() - fetch result data from last position from specific request defined by ID
- delete() - delete request with specific ID
- deleteSession() - delete all requests created during the actual SecM session.

Simple requests (without ODF)

- getInterfaceVersion() - get the version of the interface

Special requests used by CMP/UC

- getFullyQualifiedNumber() - transform a number into E.164 format
- getManagedNodes() - get the list of managed nodes (related only to Manager)
- getVersion() - get the version description of the current system
- getSNMPCommunities() - get the list of configured SNMP communities
- setSNMPCommunity() - set the SNMP community
- setSNMPTrapDestination() - set the SNMP trap destination
- prepareDeltaSynch() - prepare data from the UC_ACTION_LOG ODF file within a specified time interval
- fetchNextResultDeltaSynch() - fetch the result from prepareDeltaSynch() operation starting from last position

9.3.1 Object Data Files info

getODFNames()

The getODFNames() operation provides a list of all ODF files which are accessible by the current user.

Input:

- no parameter

Output:

- list of strings - all available ODF files

Exception:

- XIEFailure

List of exceptions:

XIEFailure.ECode=EX_SERVER_ERROR - Error during communication with the XIE server

getODFContent()

The getODFContent() operation provides the description of a specific ODF file. The description contains the name and type of every column of the ODF file.

Input:

- odfName - string - name of ODF

Output:

- list of ColumnRecord

Exception:

- XIEFailure

List of exceptions:

XIEFailure.ECode=EX_SERVER_ERROR - Error during communication with the XIE server

9.3.2 Object Data Files request control

processData()

The processData() operation starts processing of the requested query/queries on the XIE server. The format of the query is based on the format of the requests to the XIE line interface. The query string must be inserted into the request type structure.

Additional parameters of the request are specified in the Request structure to control processing:

- export/import request - set via ActionType
- synchronous/asynchronous execution - set via SyncType

Only one export query (SELECT/SELECT_UPDATES) can be executed in one request to the XIE Web Service. One or more import queries (INSERT/UPDATE/DELETE) can be executed in one request.

The count of import queries is limited to 1000 in one request.

Input:

- request - Request - data type which contains the query prepared for processing on the XIE server

Output:

- integer - ID of the executed request. This ID is used in consequent requests to get the status of request, fetch output etc.

Exception:

- XIEFailure

List of exceptions:

XIEFailure.ECode=EX_SERVER_ERROR - Error during communication with the XIE server

XIEFailure.ECode=EX_SERVER_ERROR - Too many queries are requested.

getStatus()

The getStatus() operation provides the status of the request identified by ID.

Input:

- requestID - integer - ID of the request

Output:

- ResultStatus - type representing the status of the request

Exception:

- XIEFailure

List of exceptions:

XIEFailure.ECode=EX_SERVER_ERROR - Error during communication with the XIE server

XIEFailure.ECode=EX_JOB_DOESNT_EXIST - Unable to find the request ID in the cache

fetch()

The fetch() operation reads the output from the processed request. The request is identified by the ID. The count of rows to read is specified by rowCount. If rowCount = -1, then all rows are read.

The output is read always from the beginning of the processed request.

Input:

- requestID - integer - ID of request
- rowCount - integer - count of rows to read

Output:

- ResultData - type representing the output from the processed XIE request

Exception:

- XIEFailure

List of exceptions:

XIEFailure.ECode=EX_SERVER_ERROR - Error during communication with the XIE server

XIEFailure.ECode=EX_JOB_DOESNT_EXIST - Unable to find the request ID in the cache

fetchNext()

The fetchNext() operation reads the output from the processed request. The request is identified by the ID. The count of rows to read is specified by rowCount. If rowCount = -1, then all rows are read.

The output is read from the last position.

Input:

- requestID - integer - ID of request
- rowCount - integer - count of rows to read

Output:

- ResultData - type representing the output from the processed XIE request

Exception:

- XIEFailure

List of exceptions:

XIEFailure.ECode=EX_SERVER_ERROR - Error during communication with the XIE server

XIEFailure.ECode=EX_JOB_DOESNT_EXIST - Unable to find the request ID in the cache

delete()

The delete() operation informs the XIE server to delete the request with the specified ID.

Input:

- requestID - integer - ID of request

Output:

- no parameter

Exception:

- XIEFailure

List of exceptions:

XIEFailure.ECode=EX_SERVER_ERROR - Error during communication with the XIE server

XIEFailure.ECode=EX_JOB_DOESNT_EXIST - Unable to find the request ID in the cache

deleteSession()

The deleteSession() operation is informative for the XIE server. The XIE server deletes all requests and all cached results related to this session opened via Security Management.

Input:

- no parameter

Output:

- no parameter

Exception:

- XIEFailure

List of exceptions:

XIEFailure.ECode=EX_SERVER_ERROR - Error during communication with the XIE server

9.3.3 Simple requests (without ODF)

getInterfaceVersion()

The getInterfaceVersion() operation provides the internal version of the interface.

Input:

- no parameter

Output:

- string - internal version of the interface

9.3.4 Special requests used by CMP/UC

getFullyQualifiedNumber()

The getFullyQualifiedNumber() operation converts a number into E.164 format. The operation is used by UC and CMP.

Input:

- nRequest - GetFullyQualifiedNumberRequest

Output:

- GetFullyQualifiedNumberResult

getManagedNodes()

The getManagedNodes() operation provides the list of Managed nodes (Assistants). The operation is related only for the Manager and is used by CMP.

Input:

- filter - string ("all" or "UC")

Output:

- GetManagedNodesResult - list of GetManagedNodesResultRecord

The GetManagedNodesResultRecord type contains NodeName, NodeIP, NodeMajorVersion and CSTA_IP properties. An exact description of GetManagedNodesResultRecord is part of the WSDL.

getVersion()

The getVersion() operation provides the version information of the current system. The operation is used by CMP.

Input:

- no parameter

Output:

- GetVersionResult type

The GetVersionResult type contains NodeSystemType, NodeMajorVersion and NodeSWRMVersion properties where NodeSystemType can be Manager or Assistant. Exact description GetVersionResult is part of the WSDL.

getSNMPCommunities()

The getSNMPCommunities() operation provides list of SNMP communities defined on current system. The operation is used by CMP.

Input:

- no parameter

Output:

- GetSNMPCommunitiesResult - list of SNMPCommunityRecord

The SNMPCommunityRecord type contains Name and Access (read or write) properties. An exact description of the SNMPCommunityRecord is part of the WSDL.

setSNMPCommunity()

The setSNMPCommunity() operation configures the SNMP community/s on an OpenScape system. The operation is used by CMP.

Input:

- SetSNMPCommunityRequest type - list of SNMPCommunityRecord

Output:

- SetSNMPCommunityResult type

The SNMPCommunityRecord type contains Name and Access (read or write) properties. An exact description of SNMPCommunityRecord is part of the WSDL.

The SetSNMPCommunityResult type contains the result property which is of SNMPResultCodeStruct type. An exact description of SetSNMPCommunityResult and SNMPResultCodeStruct is part of the WSDL.

setSNMPTrapDestination()

The setSNMPTrapDestination() operation configures the SNMP trap destination/s on an OpenScape system. The operation is used by CMP.

Input:

- SetSNMPTrapDestinationRequest type - list of SNMPTrapDestination

Output:

- SetSNMPTrapDestinationResult type

The SNMPTrapDestination type contains Community, IPAddress, Mask properties. An exact description of SNMPTrapDestination is part of the WSDL.

The SetSNMPTrapDestinationResult type contains the result property which is of SNMPResultCodeStruct type. An exact description of SetSNMPTrapDestinationResult and SNMPResultCodeStruct is part of the WSDL.

prepareDeltaSynch()

The prepareDeltaSynch() operation processes records from the UC_ACTION_LOG ODF file within a specified time interval. Records are inserted into the cache and can be loaded by the client with a fetchNextResultDeltaSynch() operation. The operation is used by UC.

Input:

- beginDate - Data start of interval
- endDate - Date - end of interval

Output:

- DeltaSynchResult type

The DeltaSynchResult type contains the requestID and countOfRecords properties. An exact description of DeltaSynchResult is part of the WSDL.

requestID is used in further fetchNextResultDeltaSynch() operations to read data.

fetchNextResultDeltaSynch()

The fetchNextResultDeltaSynch() operation reads output which was processed by a prepareDeltaSynch() operation. Each record in the output is mapped into a UCActionLogRecord type to simplify access from the client application. The operation is used by UC.

Input:

- requestID - integer - request ID
- recordCount - integer - record count to read

Output:

- GetUCActionLogResult type - list of UCActionLogRecord

The UCActionLogRecord type is described in WSDL.

9.3.5 Simple Types

9.3.5.1 ActionType

ActionType is an enumeration type. Valid values are:

- EXPORT_ACTION
- IMPORT_ACTION

ActionType is used in processData() operations to distinguish whether the request to the XIE server will import (INSERT/UPDATE/DELETE) or export (SELECT/SELECT_UPDATES) data.

9.3.5.2 ErrCode

ErrCode is an enumeration type. Valid values are:

- EX_OK - system is working fine
- EX_FILE_NOT_FOUND - ODF file was not found
- EX_WRONG_FORMAT - wrong format if input
- EX_JOB_DOESNT_EXIST - request doesn"t exist on server
- EX_SERVER_ERROR - server error, check long description
- EX_DATA_NOT_PREPARED - process was not finished yet
- EX_ERROR - error

ErrCode type is part of an XIEFailure type which describes exceptions/errors reported by the server. ErrCode describes the reason of the failure as a code value. A longer description is available in the error message.

9.3.5.3 RecordType

RecordType is an enumeration type. Valid values are:

- SELECT
- SELECT_UPDATES
- INSERT
- UPDATE
- DELETE

RecordType defines type of request for XIE server. RecordType in used in processData() operation.

Enumeration values inherit from request format defined in XIE file interface.

9.3.5.4 Status

- Status is an enumeration type. Valid values are:
- NOT_PROCESSED
- INPROGRESS
- DONE
- FAILED

Status describes status of processing of request to XIE server. Status is used in getStatus operation and operations related to fetch data from server.

9.3.5.5 SyncType

SyncType is an enumeration type. Valid values are:

- SYNC
- ASYNC

This type is used in Request type to distinguish whether a processData() operation will wait until the entire request is processed (SYNC) or whether it will finish immediately (ASYNC). A consequent operation getStatus() must be then executed to periodically check the status of the request.

9.3.6 Complex Types

9.3.6.1 ColumnRecord

Schema of type:

Table 8: ColumnRecord

Property name	Type	Description
name	string	column name
type	string	type of column

ColumnRecord represents one column in the ODF file. ColumnRecord type is used in getODFContent to describe the format of the ODF file.

9.3.6.2 GetFullyQualifiedNumberRequest

Schema of type:

Table 9: GetFullyQualifiedNumberRequest

Property name	Type	Description
OperatorId	string, length 12	operator ID
InputFQN	string, length 36	input FQN
DialedDigits	string, length 12	dialed digits

GetFullyQualifiedNumberRequest is used by getFullyQualifiedNumber() operation as input.

9.3.6.3 GetFullyQualifiedNumberResult

Schema of type:

Table 10: GetFullyQualifiedNumberResult

Property name	Type	Description
Result	ResultCodeStruct	
FullyQualifiedNumber	string	

GetFullyQualifiedNumberResult is used by the getFullyQualifiedNumber() operation as output.

9.3.6.4 RecordStatus

Schema of type:

Table 11: RecordStatus

Property name	Type	Description
st	Status	
status		
error_code	integer	error code
error_description	string	error description

This type represents the return value from each query on the XIE server. Check these properties to find out whether the was correctly processed on the XIE server. error_code and error_description contain the error information for the XIE server.

9.3.6.5 RequestDataRecord

Schema of type:

Table 12: RequestDataRecord

Property name	Type	Description
queryNumber	integer	identifier, starting from 1
queryType	RecordType	query type
queryString	string	XIE query statement, format of the query must be same as the format of the XIE file interface

RequestDataRecord represents one query in Request type.

The queryString has to be in a similar format as requests for File Interface.

The queryString can consist of the following based on chosen queryType (RecordType):

Command for DELETE queryType

DELETE;odf_name;value1|....|valuen|

Deletes a transferred record from OpenScape 4000 Manager/Assistant. You only need to specify the value1 field, i.e. the unique_key to identify the object to be deleted.

Command for INSERT queryType

INSERT;*odf_name;value1*|....|*valuen*|

Inserts a transferred record into OpenScape 4000 Manager/Assistant. All fields must always be specified. The first value can be empty. If the first value is specified, it has to be an unique_key word.

Command for UPDATE queryType

UPDATE;odf_name;new_value1|...|new_valuen|

Replaces the value in the old record with the value in the new record. In order to use this operation, it must not be possible for an update request to change the values of the key attributes. The key attributes depend on the ODF. A unique internal key attribute (referred to as the **unique_key**) is maintained for each object in the OpenScape 4000 Manager/Assistant database. This internal value is specified as the first value in the standard ODFs in Chapter 10, "Description of ODFs". You must always specify all fields for the update command. If no value is specified, the associated attributes are assigned the value ZERO. The first value must be a unique_key word.

Command for SELECT queryType

SELECT field1, field2,... FROM odf_name WHERE condition

Supplies all records which meet the specified condition. Only one condition can be specified here. The condition syntax is the same as the SQL syntax with one restriction: sub-queries are not allowed

Command for SELECT_UPDATES queryType

SELECT_UPDATES field1, field2,... FROM odf_name WHERE

LAST_UPDATE > time

Supplies all records which meet the specified condition (no further conditions may be specified!). You can replace the operator > in the WHERE clause with any other compare operator, The time of the last record changed is stored in LAST_UPDATE. You must specify time in the Informix date format DATETIME to FRACTION(2) (e.g. "2000-12-31 19:30:20.00"). "AND" links in the "WHERE" clause with the LAST_UPDATE attribute can only be entered.

9.3.6.6 Request

Schema of type:

Table 13: Request

Property name	Type	Description
aType	ActionType	import/export request
sType	SyncType	request is synchronous/ asynchronous
requestDataRecords	list of RequestDataRecord	

The request type represents the input for a processData() operation. Check the description of the inner types to get an overview.

Restriction: If aType is EXPORT_ACTION, there can be only one RequestDataRecord in the Request.

9.3.6.7 ResultCodeStruct

Schema of type:

Table 14: ResultCodeStruct

Property name	Type	Description
ResultCode	integer	
ResultMsgNumber	integer	
ResultTOY	dateTime	
JobId	string	
ResultText1	string	
ResultText2	string	
ResultText3	string	
ResultText4	string	
ResultText5	string	

ResultCodeStruct is used by the getFullyQualifiedNumber() operation.

9.3.6.8 ResultData

Schema of type:

Table 15: ResultData

Property name	Type	Description
st	ResultStatus	status
resultDataRecords	list of ResultDataRecord	processed data

ResultData type represents the output from fetch() and fetchNext() operations. Check the description of the inner types to get an overview.

9.3.6.9 ResultDataRecord

Schema of type:

Table 16: ResultDataRecord

Property name	Type	Description
queryNumber	integer	identification of query inside result, its same number as queryNumber in RequestDataRecord
type	RecordType	type of query
st	RecordStatus	status or processed query
countOfRecords	integer	count of records in output
records	list of string	output from query

ResultDataRecord represents the output from one processed query from the request. The record"s property contains the output from the processed query.

If the ActionType of request was EXPORT_ACTION, the record"s property contains a list of string with output. The values in the output are separated by a "|" pipe character. If the ActionType of the request was IMPORT_ACTION, the record"s property is empty.

9.3.6.10 ResultStatus

Schema of type:

Table 17: ResultStatus

Property name	Type	Description
st	Status	constant describing status
processedRequestRecords	integer	count of processed records
requestRecords	integer	total count of records

ResultStatus represents the processing status of a request. The cpecific request is identified by the ID which is passed to the getStatus() operation.

9.3.6.11 XIEFailure

XIEFailure type represents the errors/exceptions which are reported by the XIE Web Service or errors/exceptions which are propagated from the XIE server to the system. The type always contains the error code and a textual description.

Schema of type:

Table 18: XIEFailure

Property name	Type	Description
ECode	ErrCode	error code

Property name	Type	Description
message	string	long error description

9.4 Client demo example

Developers who are not familiar with WSDL can use the Demo client downloadable from OpenScape 4000 Manager XIE web service page as an example:

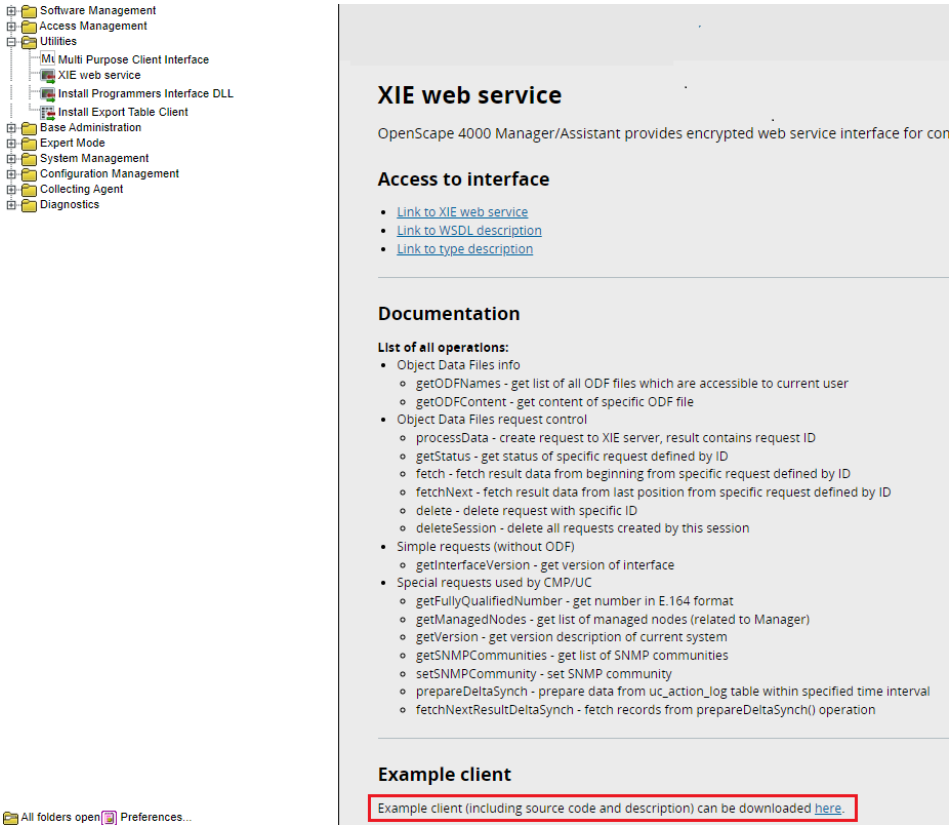

Figure 36: Client Demo Example

This example includes a simple client written as a project in NetBeans 7.2 with EE extension under JDK 1.6.

9.5 How-to write a Client

9.5.1 Requirements

- Java development - SecMCJ.jar library to open a secure session to the OpenScape system
- C++ development - secm.dll library to open a session to the OpenScape system

- WSDL description of the interface for the generation of the client code. How to download this is described in Chapter 10, "Description of ODFs".

9.5.2 How to open a secure session on the OpenScape system

This case describes the Java case only.

SecMCJ.jar can be downloaded from the system via the following URL:

```
https://<IP>/public/classes/SecMCJ.jar
```

Class de.siemens.icn.hipath.unity.as.general.secm.SecMCJSession from SecMCJ.jar library can be used to open a secure session. The session cookie identifies the session on the OpenScape system in all subsequent requests.

```
CookieManager manager = new CookieManager();
CookieHandler.setDefault(manager);
CookieStore cookieJar = manager.getCookieStore();

SecMCJSession sess = new SecMCJSession(ip, 443, user, password, "en");
sess.open();
String cookie = sess.getCookie();

String[] ccc = cookie.split("=");
HttpCookie hCookie = new HttpCookie(ccc[0], ccc[1]);
cookieJar.add(URI.create(ip), hCookie);
```

9.5.3 Sequence of operations in user scenario

In this scenario, the user requests data processing and read of response.

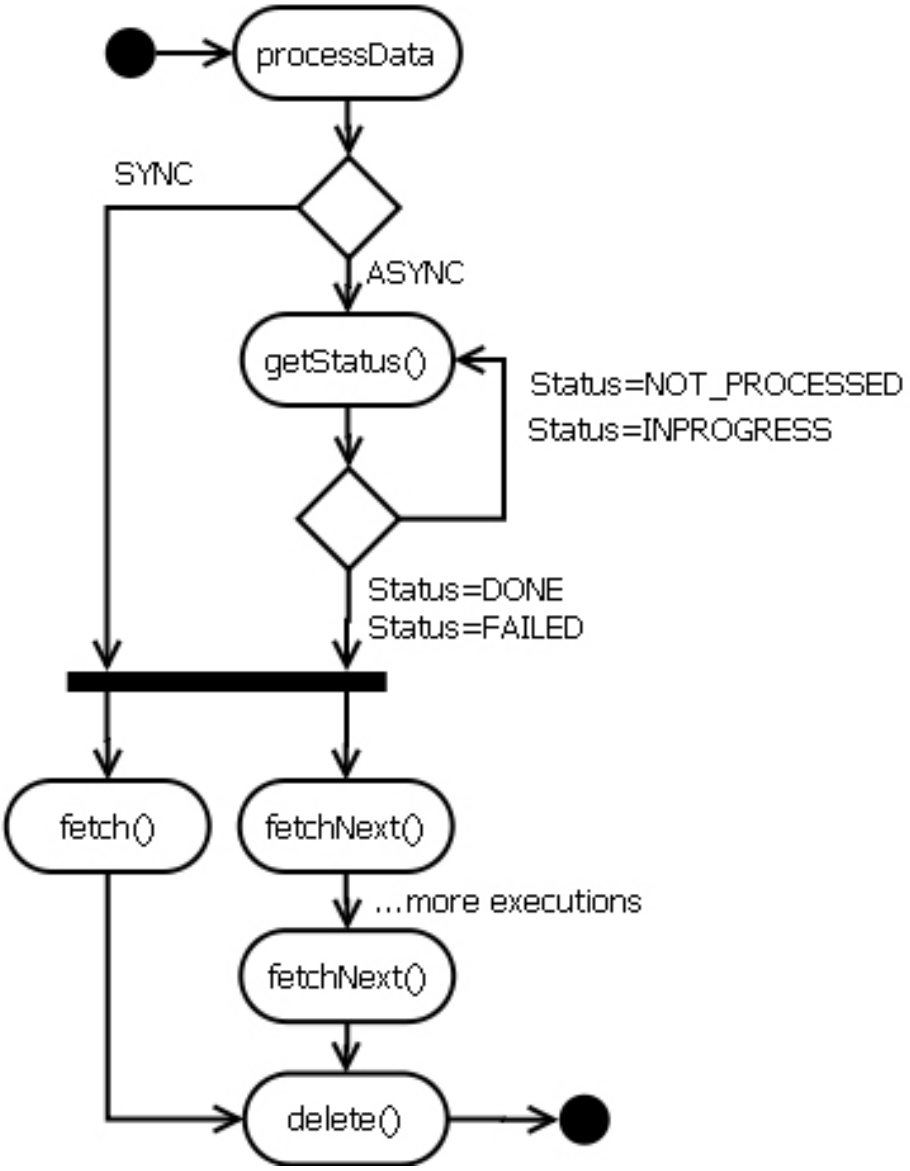

Figure 37: Sequence of operations in user scenario

1) Authentication via OpenScape Security Management is required. Open a web session on the system via SecMCJ.jar library (class de.siemens.icn.hipath.unity.as.general.secm.SecMCJSession).

2) Send data for XIE processing via the processData(request) operation. The result contains the ID of the request. If the SyncType in the request is set to SYNC, the request finishes after the data have been processed, If the SyncType in the request is set to ASYNC, the data are processed asynchronously and the request finishes immediately.

3) Check the status of XIE processing with the getStatus(ID) operation. This step is only needed in case of asynchronous processing.

4) If the status is INPROGRESS or NOT_PROCESSED, repeat the request.

5) If the status is DONE or FAILED, fetch the data from the request with fetch(ID, rowCount) or fetchNext(ID, rowCount).

6) Execute the delete(ID) operation when all data are read. This operation deletes the cached result on the XIE server.
7) Alternative step: Execute deleteSession(). This operation will delete the cached results for all requests related to this Security Management session (identified by cookie).
8) Final step: Close the Security Management session (class de.siemens.icn.hipath.unity.as.general.secm.SecMCJSession).

9.5.4 Example of usage of processData() operation

9.5.4.1 Select Statement

Select all records from DEVCOMB table.

```
int rID = -1;
RequestDataRecord rdrIt;

 //Construct Request
request = new Request();
 //Fill request based on GUI
request.setAType(ActionType.EXPORT_ACTION);
request.setSType(SyncType.ASYNC);

 //Fill queries into RequestDataRecord type
rdrIt = new RequestDataRecord();
rdrIt.setQueryNumber(1);
rdrIt.setQueryType(RecordType.SELECT);
rdrIt.setQueryString("select * from DEVCOMB");
request.getRequestDataRecords().add(rdrIt);

 try {
   //Send request to web service + get ID of request on server side
   rID = port.processData(request);
} catch (XIEFailure_Exception ex) {
 //Error handling
ex.printStackTrace();
}
```

9.5.4.2 Insert statement

Insert 2 records into PERSDAT ODF file.

```
    int rID = -1;
    RequestDataRecord rdrIt;

    //Construct Request
   request = new Request();
    //Fill request based on GUI
   request.setAType(ActionType.IMPORT_ACTION);
   request.setSType(SyncType.SYNC);

    //Fill queries into RequestDataRecord type
   rdrIt = new RequestDataRecord();
   rdrIt.setQueryNumber(1);
   rdrIt.setQueryType(RecordType.INSERT);
rdrIt.setQueryString("INSERT;PERSDAT;843|||||||||ole|OLE|ola|||||||||||||||
20130206|||||
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||000|000||||||
kkk@excha
nge.com|80726||");
   request.getRequestDataRecords().add(rdrIt);

    rdrIt = new RequestDataRecord();
   rdrIt.setQueryNumber(2);
   rdrIt.setQueryType(RecordType.INSERT);
rdrIt.setQueryString("INSERT;PERSDAT;843|||||||||ole|OLE|ola|||||||||||||||
20130206|||||
||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||||000|000||||||
kkk2@exch
```

```
ange.com|80722||");
   request.getRequestDataRecords().add(rdrIt);

    try {
      //Send request to web service + get ID of request on server side
      rID = port.processData(request);
   } catch (XIEFailure_Exception ex) {
    //Error handling
   ex.printStackTrace();
   }
```

9.5.4.3 Example of usage of getStatus() operation

```
    int requestID = <id_from_processData>;
   ResultStatus rStatus;

    try {
       //Send request to server
      rStatus = port.getStatus(requestID);
       //Parse received status
      System.out.println("processed lines: " +
       rStatus.getProcessedRequestRecords());
      System.out.println("record lines: " + rStatus.getRequestRecords());
      System.out.println("status: " + rStatus.getSt().value());

    } catch (XIEFailure_Exception ex) {
    //Error handling
   ex.printStackTrace();
   }
```

9.5.4.4 Example of usage of fetch/fetchNext operation

```
int requestID = <id_from_processData>;
int rCount = 100;
ResultData rData;
ResultDataRecord rdr;


 try {
    //Send request to web service
   rData = port.fetchNext(requestID, rCount);


    //Parse status data from response
   System.out.println("processed records: " +
   rData.getSt().getProcessedRequestRecords());
   System.out.println("request records: " +
    rData.getSt().getRequestRecords());
   System.out.println("status: " + rData.getSt().getSt().value());


    //Parse processed data from response
   int ccount = rData.getResultDataRecords().size();
   if (ccount > 0) {
      for (int i = 0; i < ccount; i++) {
         rdr = rData.getResultDataRecords().get(i);
         System.out.println("Record number: " + rdr.getQueryNumber());
         System.out.println("Type: " + rdr.getType().value());
         System.out.println("Status: " + rdr.getSt().getSt().value() + "\nError
          code:"
         + rdr.getSt().getErrorCode()
            + "\nErrorDescription: " + rdr.getSt().getErrorDescription());
         System.out.println("Anzahl Datensätze (mit SELECT/UPDATE/DELETE
          bearbeitet)="
         + rdr.getCountOfRecords());


          int ccount2 = rdr.getRecords().size();
         if (ccount2 > 0) {
           //List processed output
           for (int j = 0; j < ccount2; j++) {
             System.out.println ("Verarbeitete Ergebnisdaten:" +
```

```
          rdr.getRecords().get(j) + "\n");
            }
          } else {
            System.out.println("Kein Datensatz!!!\n");
          }
        }
    }
  } catch (XIEFailure_Exception ex) {
      //Error handling
     ex.printStackTrace();
  }
```

10 Description of ODFs

10.1 List of ODFs

- The following lists contain ODFs with the authorized operations divided into groups depending on their purpose. Within a group the list is sorted alphabetically. In the OpenScape 4000 area there are the following groups:
- OpenScape 4000 Objects - these objects are accessible by all interfaces. For most cases they are accessible from both Manager/Assistant.
- File Interface Objects - these objects are accessible only by the File Interface from both Manager/Assistant.
- Assistant Objects - these are accessible by both interfaces but only from the Assistant (in these cases they are also called non-traditional objects)
- HIM Objects - these are accessible by both interfaces but only from the Assistant.
- OpenScape/HiPath Accounting Management Objects - they are accessible from both Manager/Assistant.

If the list of import operations differ on OpenScape4000 Manager/Assistant or used interface, this is mentioned in the Relevance/Restriction column, and the object is marked in blue.

The ODFs which are deprecated and should not be used are marked in red.

10.1.1 OpenScape 4000 Objects

List of object data files accessible by all interfaces (FileInterface, WebService, Export Table Client, api2hipath, API-Class Lib) on both OpenScape 4000 Manager and OpenScape 4000 Assistant. The list also includes information about supported Export/Import operations.

Table 19: OpenScape 4000 Objects

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
ACCESS_RIGHTS	x		x	x	x	Access right
ACCESS_RIGHTS_SUB	x		x	x	x	Access right
ACTION_CONTROL	x					Progress of inside Confi Managemer
ALARMS	x					List of alarm
APPEARANCE	x					Appearance PRIM keys
APPEARANCE_ON	x					Same as AP
ATM_BOARD	x					Atm board

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
BERUM	x					Active class of ser changeover
BUEND	x					Trunk Groups
BUILDING	x		x	x	x	Building for Syste Configuration Mar
BUTTON_INDIV	x					Individual Key Lay
BUTTON_STANDARD	x					Standard Key Lay
BUTTON_UPD	x	x				For update operat Layout
CFW	x		x	x	x	Call forwarding da
CHDMAIN	x					List of Managed S
CHESE	x		x		x	Executive Secreta
CHESECOMB	x					Executive Secreta members combina
CHESE_SUB	x		x		x	Members of Exec Secretary Group
CHESE_TEAM	x					Executive Secreta Arrangements in r team view
CMI_BASE_STAT	x					Cordless base sta
CMI_BOARD	x					Cordless board
CMI_BOARD_SUB	x					Cordless board su
CMI_QUER	x					Cordless object
CMI_SYSTEM	x					Global Cordless-E
CMI_SYSTEM_SUB	x					Cordless object
CODED_VAL	x					Description of cod in Configuration M
COEX	x					External Class of

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
COMPANY_CODE	x					Company C
COMPORG	x					Company/O join
COS	x		x	x	x	Class of Ser
COSXT	x					Times for cla switchover
CTRYNUM	x		x	x	x	Country cod Configuratio
DEVCOMB	x					Device Com
DEVCONST	x					Device Com
DIMSU	x					Dimensionin switching un
DISPLAY_NAME	x		x	x	x	Display nam organization
DOMAIN	x		x	x	x	Domain data Managemen
DOWNLOAD_CLASS	x					Download C Configuratio
DOWNLOAD_PARAMS	x					Download P Configuratio
DOWNLOAD_SUPP	x					Download su Configuratio
DYNAMIC_KEYDAT	x					Dynamic Ke
E164_NUM	x					Port/Numbe
ERRORS	x					Fault Messa history
FIRMDAT	x		x	x	x	Company da Configuratio
GRP_PBOOK	x					Group Phon
GRP_PBOOK_BRDCAST	x		x	x	x	Group Phon
GRP_PBOOK_SUB	x		x	x	x	Members of group
HUNTGRP	x					Hunt Group

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
HUNTGRP_AC	x					Hunt Group Acces
HUNTGRP_DPLN_CPS	x					Hunt Group Call F State
HUNTGRP_SERVICE	x		x	x	x	Hunt Group Servic
HUNTGRP_SUB	x		x	x	x	Members of Hunt
KNPRE	x					Used to get the di (Node Prefix Table
KZNR	x		x	x	x	Speed dialing
KZNR_LIST	x		x	x	x	Speed dial list
LBER	x					Least Cost Routin
LCOS_DATA	x					Least Cost Routin Service - Data
LCOS_VOICE	x					Least Cost Routin Service - Voice
LCR	x					Least Cost Routin
LCRAUTH	x					Least Cost Routin Authorizations
LCRDPLN	x					Least Cost Routin Administration
LCRDPLN_SUB	x					Elements of Least Routing Dial Plan
LCRRULE	x					LCR Dialing rule
LCRRULE_SUB	x					LCR Dialing Rule
LCRSCHED	x					Time dependent r selection control
LCRZONE	x					LCR zone informa
LOCDAT	x		x	x	x	Location
LRTG	x					LCR lrt group
LRTG_SUB	x					Members of LCR
NUMBERING_PLAN	x					Numbering plan fo
ORDERS	x					Orders table for b
ORG	x		x	x	x	Organization data
PEN	x		x	x	x	PEN data

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
PERSDAT	x	x	x	x	x	Personal (su
PERSDAT_VNR	x					Personal da VNR
PERSPORT	x					Personal da join betwee display_nam
PICKUPGRP	x		x		x	Use to creat
PICKUP_LINK	x					Pickup link c
PICKUP_SUB	x		x		x	Use to add r PickupGrou
PIN	x	x	x	x	x	Use to mana
PINCHECK	x					PIN check
PINCLASS	x					PIN class
PINGENERAL	x					PIN general
PINPERS	x					Persdat dom between per
PORT	x	x	x	x	x	Use to mana
PORTPERS	x					PORT domi port/persdat
PRIV_DIAL_NUMBER	x		x	x	x	Use to mana numbers
PRIV_DIAL_NUM_RA	x					Private dial
PRIV_DIAL_NUM_ST	x					Private dial
PRIV_DIAL_NUM_WV	x					Private dial
ROUTE_INFO_WN	x					Route info (s
ROUTE_INFO_WT	x		x	x	x	Route info (l
SBS_PORT	x	x	x	x	x	Used to mar for SBS (wit names)
SHELF_DATA	x					Shelf data
SPEC_SUBSCRIBER	x					Special Sub
STATION_RANGE	x		x	x	x	Use to mana Range
STATION_XREF	x					Station cros

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
SWITCH	x		x	x	x	Use to manager S
TRANS_SERVICE						Translation Servic translate number i
UC_ACTION_LOG	x					Log of actions exe UC
UPD_PERSDAT	x					Select Updates Ta Personal data
UPD_PIN	x					Select Updates Ta data
UPD_PORT	x					Select Updates Ta PORT data
VBZ	x					Traffic restriction
VBZGR	x					Traffic restriction g
VBZ_AUSWEIS	x					Traffic restriction
VMSCOS	x					Voice mail class o
VOICE_MAIL	x					Voice mail
XPR_DATA	x					View persdat/disp join for expressior

10.1.2 File Interface Objects

List of the object data files accessible by **File Interface** and **WebService** only. These object data files are available on both Manager and Assistant. The list also includes information about supported Export/Import operations.

Table 20: File Interface Objects

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
AM_VNR_PERS	x	x				Personal Data wi VNR extensions
AM_VNR_PIN	x	x				PIN Data with VN extensions
AM_VNR_SWITCHDATA	x					Switch data with extensions
BUILDIMP	x		x	x	x	For Building impo

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
COMPIMP	x		x	x	x	For Compa
COSDATA	x					Class of Se View
CTRYCODES	x		x	x	x	Country Co
DEVCOM	x		x	x	x	For Devcon
DOMAINDATA	x					Domain Da
EXTENSIONS	x					Private Dial view
GRP_PBOOK_DATA	x		x	x	x	For import Phone Boo
HUNT	x					For view th between hu hunt group huntgrp ser
LOCIMP	x		x	x	x	For Locatio
ORGIMP	x		x	x	x	For Organiz Import
PENDATA	x					PEN Data
PERS	x	x	x	x	x	For Person import
PERS_VNR	x					Personal D VNR extens
PICKUP	x					Pickup exp
PINDATA	x	x	x	x	x	For PIN dat
PORT_DISP	x					Join betwee and DISPL
PORTDATA	x	x				Port data vi
PORTUPD	x		x	x	x	For update DISPLAY n
SBS_PO_PER	x	x				Personal D port join for
SNMP_TRAP_DEST	x					SNMP Trap Destination view
SNMPV1_USER	x					SNMPv1co settings vie

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
SWITCHDATA	x					Switch data expo
UPD_BUTTON_INDIV				x		For update indivi key layout only
V_PERS	x	x	x	x	x	For export/import Personal Data int virtual switch
V_PERS_VNR	x					View of Personal Data in Virtual Sw enhanced by VNI
V_PINDATA	x	x	x	x	x	For export/import PIN Data into virt switch

10.1.3 Assistant Objects

List of so called **non-traditional** object data files. These object data files are available on the Assistant only. They are accessible by all interfaces (File Interface, WebService, Export Table Client, api2hipath, API-Class Lib).

Table 21: Assistant Objects

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
ACD_AGENT	x		x	x	x	ACD Agent
ACD_ANNOUNC	x					ACD Announcement Device
ACD_GROUP	x					ACD Group
ACD_MUSIC	x					ACD Music Device
ACD_NUMBER	x					ACD Number
ACD_RCG	x					ACD Route Contro Group
ACD_RCG_SUB	x					ACD Route Contro Member
ACD_RTG_TABLE	x					ACD Routing Table
ACD_RTG_TABLE_SUB	x					ACD Routing Table Member
ACD_SHIFT_SET	x					ACD Shift Set
ACD_SHIFT_SET_SUB	x					ACD Shift Set Member

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
ACD_SYS_PARAM	x					ACD System Parameters
BOARD	x					Board
UBGDAT	x					Ubgdat
FIREWALL	x					Firewall
FIREMAC	x					Firemac
IPROUTING	x					IP Routing
REMOTE_TERM	x					Remote Term
NETWORK	x					Network Inter
COP	x					Class of Parameters
COT	x					Class of Trun
PM_RTG_ACI	x					RICHT Phone Routing - Atte Console Ove
PM_RTG_MWI	x					RICHT Phone Routing - Mes Waiting Index
PM_RTG_FAXO	x					RICHT Phone Routing - Fax Overflow

10.1.4 OpenScape/HiPath Inventory Management Objects

List of so called **OpenScape/HiPath Inventory Management** object data files. These object data files are available on Assistant only. They are accessible by all interfaces (FileInterface, WebService, Export Table Client, api2hipath, API-Class Lib).

Table 22: HIM Objects

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Descriptio
BOARD_TYPE	x					Board Typ
CABINET	x					Cabinet da
CABINET_1	x					Cabinet da
CABINET_VIEW	x					Cabinet da
CENTRAL_SIU	x					Central Si Unit data

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
CENTRAL_SIU_VIEW	x					Central Signalin Unit data view
CONSY_CAB	x					CONSY Cabine
CONSY_FRAME	x					CONSY Frame
COS_DESCRIPTION	x					COS Descriptio
CUST_FEATURE	x					Customer featu overview
DBC_CONF_DATA	x					Database Confi data
DBC_HW_DATA	x					Database Confi Hardware data
DBC_SYS_DATA	x					Database Confi System data
DIMSU_VIEW	x					Memory Scaling
FEA_HEADER	x					Features overvi
FEA_SOLD	x					Bought features
FEA_USAGE	x					Used features o
HW_INFO	x					Hardware inform
LCOS_DESCRIPTION	x					LCOS Descripti
LW_ON_CB	x					Loadware Data Board
LW_ON_CENTRAL_SIU	x					Loadware Data Signaling Interfa
LW_ON_PROCESSOR	x					Loadware Data Processor
MAC_ADDRESS_VIEW	x					MAC Addresses
PATCH	x					Software Patche Information
PEN_LW	x					PEN loadware
PORT_STATION_VIEW	x					Port station view
PROJ_PLAN_DATA	x					Project Planing Information
PSIO_VIEW	x					Peripheral Seria Output data view

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Descriptio
REPLACED_AMO	x					Replaced that have since the correction
SCSI_DEV	x					SCSI devi
SCSI_DEV_VIEW	x					SCSI devi
SERIAL_LINE	x					Serial inte
SHELF	x					Configure
SHELF_VIEW	x					Configure view
SW_VERSION	x					Software
SWITCH_APS	x					Applicatio System or
SWITCH_NO	x					Switch nu
SWU_BOARD	x					SWU boa
SWU_BOARD_VIEW	x					SWU boa
SWU_N_DATA	x					SWU N da
SWU_PORT_ATTENDANT	x					SWU port
SWU_PORT_SPECIAL	x					SWU port
SWU_T_DATA	x					SWU T Da
TRUNK_VIEW	x					Trunk view
UNIX_SW	x					Software installed o
WELCOME_PAGE_VIEW	x					HIM Welc overview
Y_APS	x					System in

10.1.5 OpenScape/HiPath Accounting Management Objects

List of object data files intended for export to **Accounting Management**.

Table 23: Accounting Management Objects

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
AM_VNR_PERS	X	X				Personal (s extensions

ODF Name	SELECT	SELECT_UPDATES	INSERT	DELETE	UPDATE	Description
AM_VNR_PIN	X	X				PIN data with VN
AM_VNR_SWITCHDATA	X					Switch data with
AMW_COMPORG	X					Company/Organi
AMW_PERSDATA	X	X				Personal (subscr
AMW_PINDATA	X	X				PIN data export f
AMW_SWITCHDATA	X					Switch data expo

10.2 Detailed Description of ODFs

The following part includes detailed descriptions for each ODF containing the objects and the associated attributes.

Every ODF description contains 3 columns, where the first column contains the field name, the second contains the field type, and the third contains a comment.

The comment may contain:

- a hint that the field contains a required value (only in case of an ODF with import function)
- a comment concerning the content of the field
- one or more of the three abbreviations: **swu** = switch-relevant, **key** = key value, and **coded** = coded value.
- additional information about possible codes.

10.2.1 OpenScape 4000 Objects - Detailed Description

10.2.1.1 ACTION_CONTROL

Table 24: ACTION_CONTROL

ACTION_CONTROL		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
action_user	char(32)	action_user
action_id	int	action_id, key
count_total	int	count_total
count_ready	int	count_ready

ACTION_CONTROL		
:EXPORT:SELECT		
log_status	char(1)	log_status
operation	char(3)	operation, coded (see Operations for action_protocol)
object_class	char(3)	object_class, coded (see Object_type for action_protocol 600-630)
attributes	char(20)	attributes
child_count	int	child_count
child_complete	int	child_complete
parent_action_id	int	parent_action_id
action_category	int	action_category
message_no	int	message_num
str1	char(16)	string1
str2	char(32)	string2
str3	char(32)	string3
str4	char(16)	string4
int5	int	int5
order_id	char(12)	order_id, key
switch_name	char(4)	switch_name
extension	char(12)	extension
number	char(12)	number
name	char(50)	name
action_status	char(3)	action_status, coded(see Error states (for action protocol))
action_error	char(3)	action_error, coded(see Error states (for action protocol))
act_time_create	char(19)	act_time_create
act_time_update	char(19)	Modification Time

10.2.1.2 ALARMS

Table 25: ALARMS

ALARMS		
:EXPORT:SELECT		
al_serial	int	serial
pabx_id	int	identification of pabx
dev_cl_gr	int	device group
dev_class	int	device class
state	int	state of the alarm
priority	int	priority of alarm
al_date	int	alarm date
msg_lauf	char(5)	alarm message sequence number
devcl_str	char(22)	device class string
abs_mod	char(5)	server id
on_off_ind	int	on/off indication
tim_dat	int	creation time
amode	int	amode
ack	int	acknowledged
ack_tim	int	acknowledged time
proc_state	int	proc state
due_date	int	due date
responsible_name	char(22)	responsible name
ack_by	char(9)	ack by
tim_old	int	time old
hts_flag	int	hts flag
server_time	int	server time

10.2.1.3 APPEARANCE

Table 26: APPEARANCE

APPEARANCE		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name, key
extension	char(12)	extension, swu, key
key_number	char(3)	number, swu, key
key_function	char(3)	key_function, swu, coded (see key functions (850-899), key functions (920-949))
key_destination	char(22)	key_destination, swu
key_protect	char(3)	Rejection, swu, coded (000=no, 001=yes)
ring_opt	char(3)	Ring Option, swu, coded
aics	char(3)	Automatic call signaling, swu, coded
key_direct	char(3)	Voice Calling, swu, coded

10.2.1.4 APPEARANCE_ON

Table 27: APPEARANCE_ON

APPEARANCE_ON		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name, key
extension	char(12)	extension, swu, key
key_number	char(3)	number, swu, key
key_function	char(3)	key_function, swu, coded (see key fu functions (920-949))
key_destination	char(22)	key_destination, swu

APPEARANCE_ON		
:EXPORT:SELECT		
key_protect	char(3)	Rejection, swu, coded (000=no, 001=yes)
ring_opt	char(3)	Ring Option, swu, coded
aics	char(3)	Automatic call signaling, swu, coded
key_direct	char(3)	Voice Calling, swu, coded

10.2.1.5 ATM_BOARD

Table 28: ATM_BOARD

ATM_BOARD		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain, swu
switch_name	char(4)	switch_name, key
card_typ	char(3)	STMA Type, swu, coded
slot	char(9)	pen, swu, key
vpi	char(1)	VPI, swu, key
vci_min	char(5)	vci_min, swu
vci_max	char(5)	vci_max, swu

10.2.1.6 BERUM

Table 29: BERUM

BERUM		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
info	char(300)	info
cosx	char(2)	cos switchover group, swu, key

BERUM		
:EXPORT:SELECT		
cosact	char(1)	active cos, swu

10.2.1.7 BUEND

Table 30: BUEND

BUEND		
:EXPORT:SELECT		
unique_key	int	unique_key
buname	char(18)	Trunk group name, swu
trunk_group	char(3)	Trunk group number, swu, key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
dest_switch	char(4)	Destination system
switch_node	char(10)	nodenum
dest_node	char(10)	Destination node number
trunk_group_type	char(3)	Trunk group type, swu, coded (see buend, trunk_group_type)
trunk_group_dest	char(3)	Destination trunk group, swu
num_of_trunks	char(4)	Number of trunks, swu
traffic_type	char(3)	Traffic type, coded (see buend, traffic_type)
info	char(300)	info
wabe	char(2)	DPLN, swu
lcosd	char(4)	LCOS data, swu
lcoss	char(4)	LCOS voice, swu
vbz	char(2)	ITR, swu

BUEND		
:EXPORT:SELECT		
trans_cnt	char(2)	Transit counter, swu
sat_cnt	char(3)	Satellite counter, swu, coded (see buend, sat_count)

10.2.1.8 BUILDING

Table 31: BUILDING

BUILDING		
:EXPORT:SELECT		
:IMPORT:INSERT: DELETE:UPDATE		
unique_key	int	unique_key
location	char(15)	(required) location, key
name	char(30)	(required) name, key
longname	char(25)	longname

10.2.1.9 BUTTON_INDIV

Table 32: BUTTON_INDIV

BUTTON_INDIV		
:EXPORT:SELECT		
unique_key	int	
domain	char(8)	
switch_name	char(4)	key
extension	char(12)	extension, swu, key
devconame	char(20)	Device combination, key
num_key_lay	char(3)	swu, key
digite_typ	char(3)	swu, key, coded (see *Devices 200-249, device types)

digtyp	char(3)	swu, coded (see *Devices 200-249, device types)
layout_typ	char(3)	coded(see exact device types)
layout_text	char(20)	
addon_typ	char(3)	coded(see addon type)
addon_text	char(20)	
addon_cnt	int	
info	char(50)	
key01_fun	char(3)	swu, coded (see key functions (850-899), key functions (920-949), 002=free)
key02_fun	char(3)	swu, coded (see above)
key03_fun	char(3)	swu, coded (see above)
key04_fun	char(3)	swu, coded (see above)
key05_fun	char(3)	swu, coded (see above)
key06_fun	char(3)	swu, coded (see above)
key07_fun	char(3)	swu, coded (see above)
key08_fun	char(3)	swu, coded (see above)
key09_fun	char(3)	swu, coded (see above)
key10_fun	char(3)	swu, coded (see above)
key11_fun	char(3)	swu, coded (see above)
key12_fun	char(3)	swu, coded (see above)
key13_fun	char(3)	swu, coded (see above)
key14_fun	char(3)	swu, coded (see above)
key15_fun	char(3)	swu, coded (see above)
key16_fun	char(3)	swu, coded (see above)
key17_fun	char(3)	swu, coded (see above)
key18_fun	char(3)	swu, coded (see above)
key19_fun	char(3)	swu, coded (see above)
key20_fun	char(3)	swu, coded (see above)
key21_fun	char(3)	swu, coded (see above)

key22_fun	char(3)	swu, coded (see above)
key23_fun	char(3)	swu, coded (see above)
key24_fun	char(3)	swu, coded (see above)
key25_fun	char(3)	swu, coded (see above)
key26_fun	char(3)	swu, coded (see above)
key27_fun	char(3)	swu, coded (see above)
key28_fun	char(3)	swu, coded (see above)
key29_fun	char(3)	swu, coded (see above)
key30_fun	char(3)	swu, coded (see above)
key31_fun	char(3)	swu, coded (see above)
key32_fun	char(3)	swu, coded (see above)
key33_fun	char(3)	swu, coded (see above)
key34_fun	char(3)	swu, coded (see above)
key35_fun	char(3)	swu, coded (see above)
key36_fun	char(3)	swu, coded (see above)
key37_fun	char(3)	swu, coded (see above)
key38_fun	char(3)	swu, coded (see above)
key39_fun	char(3)	swu, coded (see above)
key40_fun	char(3)	swu, coded (see above)
key41_fun	char(3)	swu, coded (see above)
key42_fun	char(3)	swu, coded (see above)
key43_fun	char(3)	swu, coded (see above)
key44_fun	char(3)	swu, coded (see above)
key45_fun	char(3)	swu, coded (see above)
key46_fun	char(3)	swu, coded (see above)
key47_fun	char(3)	swu, coded (see above)
key48_fun	char(3)	swu, coded (see above)
key49_fun	char(3)	swu, coded (see above)
key50_fun	char(3)	swu, coded (see above)
key51_fun	char(3)	swu, coded (see above)
key52_fun	char(3)	swu, coded (see above)
key53_fun	char(3)	swu, coded (see above)

key54_fun	char(3)	swu, coded (see above)
key55_fun	char(3)	swu, coded (see above)
key56_fun	char(3)	swu, coded (see above)
key57_fun	char(3)	swu, coded (see above)
key58_fun	char(3)	swu, coded (see above)
key59_fun	char(3)	swu, coded (see above)
key60_fun	char(3)	swu, coded (see above)
key61_fun	char(3)	swu, coded (see above)
key62_fun	char(3)	swu, coded (see above)
key63_fun	char(3)	swu, coded (see above)
key64_fun	char(3)	swu, coded (see above)
key65_fun	char(3)	swu, coded (see above)
key66_fun	char(3)	swu, coded (see above)
key67_fun	char(3)	swu, coded (see above)
key68_fun	char(3)	swu, coded (see above)
key69_fun	char(3)	swu, coded (see above)
key70_fun	char(3)	swu, coded (see above)
key71_fun	char(3)	swu, coded (see above)
key72_fun	char(3)	swu, coded (see above)
key73_fun	char(3)	swu, coded (see above)
key74_fun	char(3)	swu, coded (see above)
key75_fun	char(3)	swu, coded (see above)
key76_fun	char(3)	swu, coded (see above)
key77_fun	char(3)	swu, coded (see above)
key78_fun	char(3)	swu, coded (see above)
key79_fun	char(3)	swu, coded (see above)
key80_fun	char(3)	swu, coded (see above)
key81_fun	char(3)	swu, coded (see above)
key82_fun	char(3)	swu, coded (see above)
key83_fun	char(3)	swu, coded (see above)
key84_fun	char(3)	swu, coded (see above)
key85_fun	char(3)	swu, coded (see above)

key86_fun	char(3)	swu, coded (see above)
key87_fun	char(3)	swu, coded (see above)
key88_fun	char(3)	swu, coded (see above)
key89_fun	char(3)	swu, coded (see above)
key90_fun	char(3)	swu, coded (see above)
key91_fun	char(3)	swu, coded (see above)
key92_fun	char(3)	swu, coded (see above)
key93_fun	char(3)	swu, coded (see above)
key94_fun	char(3)	swu, coded (see above)
key95_fun	char(3)	swu, coded (see above)
key96_fun	char(3)	swu, coded (see above)
sync_by	char(4)	sync by
sync_time	int	sync_time
info2	char(50)	AMO description, swu

The ODF BUTTON_INDIV (described above) is available only when using the Class Library (DLL, Export Table Client, api2hipath), not when using the File Interface !

10.2.1.10 BUTTON_UPD

Table 33: BUTTON_UPD

BUTTON_UPD		Join between button_standard, port, dynamic_keydat, devconst and devtyp_relation
:EXPORT:SELECT		
:IMPORT:UPDATE		
domain	char(8)	Domain, key
switch_name	char(4)	Switch name, key
extension	char(12)	(required) extension, swu, key
key_number	char(3)	(required) key number, key, 01-32: keys of the main device, 33-224: keys of addon devices
devconame	char(20)	Device combination, key
num_key_lay	char(3)	swu, key

digite_typ	char(3)	swu, key, coded (see *Devices 200-249, device types)
digtyp	char(3)	swu, coded (see *Devices 200-249, device types)
layout_typ	char(3)	coded(see exact device types)
layout_text	char(20)	layout text, value will be ignored in case of update
addon_typ	char(3)	coded(see addon type)
addon_text	char(20)	addon text, value will be ignored in case of update
addon_cnt	int	addon cnt, value will be ignored in case of update
key_deviation	char(3)	deviation, value will be ignored in case of update
key_function	char(3)	swu, coded (refer key functions (850-899), key functions (920-949), 002=free)
key_destination	char(22)	destination number
key_protect	char(3)	swu, coded (001=yes, 000=no)
ring_opt	char(3)	swu, coded
aics	char(3)	swu, coded
key_direct	char(3)	swu, coded (001=yes, 000=no)
label_text	char(12)	swu
label_option	char(3)	swu, show extension / Anzeige des Namens, coded

The ODF BUTTON_UPD (described above) is available only when using the Class Library (DLL, Export Table Client, api2hipath), not when using the File Interface !

10.2.1.11 BUTTON_STANDARD

Table 34: BUTTON_STANDARD

BUTTON_STANDARD		
:EXPORT:SELECT		
unique_key	int	unique key
domain	char(8)	domain
switch_name	char(4)	key

num_key_lay	char(3)	swu, key
digite_typ	char(3)	swu, key, coded (see *Devices 200-249, device types)
digtyp	char(3)	swu, coded (see *Devices 200-249, device types)
layout_typ	char(3)	coded(see exact device types)
layout_text	char(30)	layout text
addon_typ	char(3)	coded(see Addon type)
addon_text	char(30)	addon text
info	char(300)	info
key01_fun	char(3)	swu, coded (see key functions (850-899), key functions (920-949), 002=free)
key02_fun	char(3)	swu, coded (see above)
key03_fun	char(3)	swu, coded (see above)
key04_fun	char(3)	swu, coded (see above)
key05_fun	char(3)	swu, coded (see above)
key06_fun	char(3)	swu, coded (see above)
key07_fun	char(3)	swu, coded (see above)
key08_fun	char(3)	swu, coded (see above)
key09_fun	char(3)	swu, coded (see above)
key10_fun	char(3)	swu, coded (see above)
key11_fun	char(3)	swu, coded (see above)

key12_fun	char(3)	swu, coded (see above)
key13_fun	char(3)	swu, coded (see above)
key14_fun	char(3)	swu, coded (see above)
key15_fun	char(3)	swu, coded (see above)
key16_fun	char(3)	swu, coded (see above)
key17_fun	char(3)	swu, coded (see above)
key18_fun	char(3)	swu, coded (see above)
key19_fun	char(3)	swu, coded (see above)
key20_fun	char(3)	swu, coded (see above)
key21_fun	char(3)	swu, coded (see above)
key22_fun	char(3)	swu, coded (see above)
key23_fun	char(3)	swu, coded (see above)
key24_fun	char(3)	swu, coded (see above)
key25_fun	char(3)	swu, coded (see above)
key26_fun	char(3)	swu, coded (see above)
key27_fun	char(3)	swu, coded (see above)
key28_fun	char(3)	swu, coded (see above)
key29_fun	char(3)	swu, coded (see above)
key30_fun	char(3)	swu, coded (see above)
key31_fun	char(3)	swu, coded (see above)

key32_fun	char(3)	swu, coded (see above)
key33_fun	char(3)	swu, coded (see above)
key34_fun	char(3)	swu, coded (see above)
key35_fun	char(3)	swu, coded (see above)
key36_fun	char(3)	swu, coded (see above)
key37_fun	char(3)	swu, coded (see above)
key38_fun	char(3)	swu, coded (see above)
key39_fun	char(3)	swu, coded (see above)
key40_fun	char(3)	swu, coded (see above)
key41_fun	char(3)	swu, coded (see above)
key42_fun	char(3)	swu, coded (see above)
key43_fun	char(3)	swu, coded (see above)
key44_fun	char(3)	swu, coded (see above)
key45_fun	char(3)	swu, coded (see above)
key46_fun	char(3)	swu, coded (see above)
key47_fun	char(3)	swu, coded (see above)
key48_fun	char(3)	swu, coded (see above)
key49_fun	char(3)	swu, coded (see above)
key50_fun	char(3)	swu, coded (see above)
key51_fun	char(3)	swu, coded (see above)

key52_fun	char(3)	swu, coded (see above)
key53_fun	char(3)	swu, coded (see above)
key54_fun	char(3)	swu, coded (see above)
key55_fun	char(3)	swu, coded (see above)
key56_fun	char(3)	swu, coded (see above)
key57_fun	char(3)	swu, coded (see above)
key58_fun	char(3)	swu, coded (see above)
key59_fun	char(3)	swu, coded (see above)
key60_fun	char(3)	swu, coded (see above)
key61_fun	char(3)	swu, coded (see above)
key62_fun	char(3)	swu, coded (see above)
key63_fun	char(3)	swu, coded (see above)
key64_fun	char(3)	swu, coded (see above)
key65_fun	char(3)	swu, coded (see above)
key66_fun	char(3)	swu, coded (see above)
key67_fun	char(3)	swu, coded (see above)
key68_fun	char(3)	swu, coded (see above)
key69_fun	char(3)	swu, coded (see above)
key70_fun	char(3)	swu, coded (see above)
key71_fun	char(3)	swu, coded (see above)

key72_fun	char(3)	swu, coded (see above)
key73_fun	char(3)	swu, coded (see above)
key74_fun	char(3)	swu, coded (see above)
key75_fun	char(3)	swu, coded (see above)
key76_fun	char(3)	swu, coded (see above)
key77_fun	char(3)	swu, coded (see above)
key78_fun	char(3)	swu, coded (see above)
key79_fun	char(3)	swu, coded (see above)
key80_fun	char(3)	swu, coded (see above)
key81_fun	char(3)	swu, coded (see above)
key82_fun	char(3)	swu, coded (see above)
key83_fun	char(3)	swu, coded (see above)
key84_fun	char(3)	swu, coded (see above)
key85_fun	char(3)	swu, coded (see above)
key86_fun	char(3)	swu, coded (see above)
key87_fun	char(3)	swu, coded (see above)
key88_fun	char(3)	swu, coded (see above)
key89_fun	char(3)	swu, coded (see above)
key90_fun	char(3)	swu, coded (see above)
key91_fun	char(3)	swu, coded (see above)

key92_fun	char(3)	swu, coded (see above)
key93_fun	char(3)	swu, coded (see above)
key94_fun	char(3)	swu, coded (see above)
key95_fun	char(3)	swu, coded (see above)
key96_fun	char(3)	swu, coded (see above)
key97_fun	char(3)	swu, coded (see above)
key98_fun	char(3)	swu, coded (see above)
key99_fun	char(3)	swu, coded (see above)
key100_fun	char(3)	swu, coded (see above)
key101_fun	char(3)	swu, coded (see above)
key102_fun	char(3)	swu, coded (see above)
key103_fun	char(3)	swu, coded (see above)
key104_fun	char(3)	swu, coded (see above)
key105_fun	char(3)	swu, coded (see above)
key106_fun	char(3)	swu, coded (see above)
key107_fun	char(3)	swu, coded (see above)
key108_fun	char(3)	swu, coded (see above)
key109_fun	char(3)	swu, coded (see above)
key110_fun	char(3)	swu, coded (see above)
key111_fun	char(3)	swu, coded (see above)

key112_fun	char(3)	swu, coded (see above)
key113_fun	char(3)	swu, coded (see above)
key114_fun	char(3)	swu, coded (see above)
key115_fun	char(3)	swu, coded (see above)
key116_fun	char(3)	swu, coded (see above)
key117_fun	char(3)	swu, coded (see above)
key118_fun	char(3)	swu, coded (see above)
key119_fun	char(3)	swu, coded (see above)
key120_fun	char(3)	swu, coded (see above)
key121_fun	char(3)	swu, coded (see above)
key122_fun	char(3)	swu, coded (see above)
sync_by	char(4)	sync by
sync_time	int	sync time
info2	char(50)	AMO description, swu

10.2.1.12 CFW

Table 35: CFW

CFW		Call Forward Data
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
extension	char(12)	extension, swu, key
dtype	char(3)	diversion type, swu, key, coded

itype	char(3)	incoming type, swu, key, coded
service	char(3)	Service indicator, swu, key, coded; i90 = voice
dest	char(22)	fcfw destination, swu
name	char(30)	Name for external destination, swu
variant	char(3)	CFW variant, swu, coded
activate	char(3)	activate, coded
template_id	int	
cfw_active	char(3)	swu, coded

10.2.1.13 CHDMAIN

Table 36: CHDMAIN

CHDMAIN		
:EXPORT:SELECT		
pabx_id	int	switch id
switch_name	char(4)	switch name
no_type	char(8)	node type
cust_server	char(4)	customer server
cust_noid	char(4)	custustomer noid
timestamp	int	timestamp
no_name	char(16)	node name
sys_type	char(10)	sys type
version	char(5)	version
description	char(40)	description
customer_specific	char(15)	customer specific
customer_contact	char(22)	customer contact
tel_no	char(22)	tel number
system_room_tel	char(22)	system room tel
street_address	char(22)	street address
location	char(22)	location
remark_cd	char(160)	remark cd

cust_name	char(30)	customer name
l_number	char(17)	l number
famos_ident	char(14)	famos ident
hicom_sw_version	char(7)	hicom software version
h_system_release	char(20)	hicom system release
hcm_amolang	char(1)	hicom amo language
unix_platform	int	unix platform
domain	char(8)	domain
node_number	char(10)	node number
remark	char(160)	remark
establ	int	establ
con_no	char(7)	con no
pos_no	char(5)	pos no
customer_contract	char(160)	customer contract
sw_lic	char(8)	sw lic
service_district	char(7)	service district
cutover	char(22)	cutover
end_warranty	char(22)	end warranty
remark_con	char(160)	remark con
line_type	char(4)	line type
famos_code	char(10)	famos code
famos_user	char(10)	famos user
rmx_ip_address	char(255)	rmx ip address
ip_address	char(255)	ip address
vpn_ip_address	char(255)	vpn ip address
remark_com	char(160)	remark com
clnr	char(22)	clnr
cb_timeout	int	cb timeout
cb_code	char(10)	cb code
msv1_proc_num	int	msv proc num

msv1_reg_num	int	msv1 reg num
hicom_name	char(6)	hicom name
chap_secret	char(30)	chap secret
ppp_password	char(8)	ppp password
c_polfl	int	c polfl
polint	int	polint
hc_equip_nr1	char(17)	hc equip nr1
hc_equip_nr2	char(17)	hc equip nr2
da_url	char(255)	da url
da_login_id	char(12)	da login id
da_login_passwd	char(24)	da login passwd
time_of_last_fm	int	time of last fm
umbrella	int	umbrella
cb_flag	char(1)	cb flag
sws_region	char(255)	sws region
hicom_status	int	hicom status
csta_ip	char(255)	
swrm_version	char(20)	
lc_id	char(3)	

10.2.1.14 CHESE

Table 37: CHESE

CHESE		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name, key
chesegrpnum	char(4)	Executive/secretary group, swu, key
info	char(300)	info
alerting_call	char(3)	alerting_call, swu, coded (see Feature 250-269)

multiple_call	char(3)	multiple_call, swu, coded (000=no, 001=yes)
tenant_group	char(20)	name of the tenant group the object belongs to Note: **Manager** only
extension_chef	char(12)	
extension_secr	char(12)	

10.2.1.15 CHESECOMB

Table 38: CHESECOMB

ODFname: **CHESECOMB** Export: **Select** Import: **-** Operations:		
Field	Data type	Description
domain	char(8)	domain name
switch	char(4)	switch name
chesegrpnum	char(4)	
info	char(50)	
alerting_call	char(3)	yes/no
multiple_call	char(3)	yes/no
extension	char(6)	extension of chief (1/2/3/4) or secretary (1/2)
chese_role	char(3)	chief or secretary
secretary	char(1)	if chese_rol = secretary, defines secretary 1 or 2, if chese_rol = chief, defines secretary 1 or 2 related to chief
remote_subscriber	char(3)	remote subscriber

10.2.1.16 CHESE_SUB

Table 39: CHESE_SUB

CHESE_SUB		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain

switch_name	char(4)	switch_name, key
chesegrpnum	char(4)	Executive/secretary group, swu, key
extension	char(12)	extension, swu, key
chese_role	char(3)	chese_role, swu, key, coded (see Feature 270-299)
secretary	char(1)	secretary, swu, key
remote_subscriber	char(3)	Remote Subscriber, swu, coded (000=no, 001=yes)
mailbox	char(3)	mailbox, swu, coded (see Feature 250-269)
position	char(1)	position
rngxfer1_keynum	char(3)	swu, coded
rngxfer2_keynum	char(3)	swu, coded
rngxfer3_keynum	char(3)	swu, coded
rngxfer4_keynum	char(3)	swu, coded
pus1_keynum	char(3)	swu, coded
pus2_keynum	char(3)	swu, coded
pus3_keynum	char(3)	swu, coded
pus4_keynum	char(3)	swu, coded
template_id	int	unique key of the record in templates table

10.2.1.17 CHESE_TEAM

Table 40: CHESE_TEAM

CHESE_TEAM		
:EXPORT:SELECT		
unique_key	int	unique_key
teamnum	char(3)	Exec./Secr. Team, swu, key
info	char(300)	info
extension_c1	char(12)	Exec. 1 extension, swu
switch_name_c1	char(4)	Exec. 1 system, swu
extension_c2	char(12)	Exec. 2 extension, swu
switch_name_c2	char(4)	Exec. 2 system, swu

extension_c3	char(12)	Exec. 3 extension, swu
switch_name_c3	char(4)	Exec. 3 system, swu
extension_c4	char(12)	Exec. 4 extension, swu
switch_name_c4	char(4)	Exec. 4 system, swu
extension_s1	char(12)	Secr. 1 extension, swu
switch_name_s1	char(4)	Secr. 1 system, swu
extension_s2	char(12)	Secr. 2 extension, swu
switch_name_s2	char(4)	Secr. 2 system, swu
extension_s3	char(12)	Secr. 3 extension, swu
switch_name_s3	char(4)	Secr. 3 system, swu
extension_s4	char(12)	Secr. 4 extension, swu
switch_name_s4	char(4)	Secr. 4 system, swu
extension_c1_s1	char(12)	Exec. 1 extension at Secr. 1, swu
extension_c2_s1	char(12)	Exec. 2 extension at Secr. 1, swu
extension_c3_s1	char(12)	Exec. 3 extension at Secr. 1, swu
extension_c4_s1	char(12)	Exec. 4 extension at Secr. 1, swu
extension_c1_s2	char(12)	Exec. 1 extension at Secr. 2, swu
extension_c2_s2	char(12)	Exec. 2 extension at Secr. 2, swu
extension_c3_s2	char(12)	Exec. 3 extension at Secr. 2, swu
extension_c4_s2	char(12)	Exec. 4 extension at Secr. 2, swu
extension_c1_s3	char(12)	Exec. 1 extension at Secr. 3, swu
extension_c2_s3	char(12)	Exec. 2 extension at Secr. 3, swu
extension_c3_s3	char(12)	Exec. 3 extension at Secr. 3, swu
extension_c4_s3	char(12)	Exec. 4 extension at Secr. 3, swu

extension_c1_s4	char(12)	Exec. 1 extension at Secr. 4, swu
extension_c2_s4	char(12)	Exec. 2 extension at Secr. 4, swu
extension_c3_s4	char(12)	Exec. 3 extension at Secr. 4, swu
extension_c4_s4	char(12)	Exec. 4 extension at Secr. 4, swu
cos_s1	char(4)	Executives COS at Secr. 1, swu
cos_s2	char(4)	Exec. lines COS at Secr. 2, swu
cos_s3	char(4)	Exec. lines COS at Secr. 3, swu
cos_s4	char(4)	Exec. lines COS at Secr. 4, swu
stbnum_s1	char(3)	Executives key layout at Secr. 1, swu
stbnum_s2	char(3)	Executives key layout at Secr. 2, swu
stbnum_s3	char(3)	Executives key layout at Secr. 3, swu
stbnum_s4	char(3)	Executives key layout at Secr. 4, swu
extension_ru_c1	char(12)	Ring transfer extension at Exec. 1, swu
extension_ru_c2	char(12)	Ring transfer extension at Exec. 2, swu
extension_ru_c3	char(12)	Ring transfer extension at Exec. 3, swu
extension_ru_c4	char(12)	Ring transfer extension at Exec. 4, swu
extension_vt_s1	char(12)	Deputy extension at Secr. 1, swu
extension_vt_s2	char(12)	Deputy extension at Secr. 2, swu
extension_vt_s3	char(12)	Deputy extension at Secr. 3, swu
extension_vt_s4	char(12)	Deputy extension at Secr. 4, swu

cnt_chef	int	Executive counter, swu
cnt_secr	int	Secretary counter, swu

10.2.1.18 CMI_BASE_STAT

Table 41: CMI_BASE_STAT

CMI_BASE_STAT		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name, key
slot	char(9)	pen, swu, key
card_type	char(3)	SLMC Type, swu, coded (see CMI board)
station_no	char(12)	Station No., swu, key
port_1	char(2)	Port no. 1, swu
port_2	char(2)	Port no. 2, swu
port_3	char(2)	Port no. 3, swu
action	char(3)	operation, swu, coded
dect_freq_1	char(3)	Frequency status 1, swu, (000=no, 001=yes)
dect_freq_2	char(3)	Frequency status 2, swu, (000=no, 001=yes)
dect_freq_3	char(3)	Frequency status 3, swu, (000=no, 001=yes)
dect_freq_4	char(3)	Frequency status 4, swu, (000=no, 001=yes)
dect_freq_5	char(3)	Frequency status 5, swu, (000=no, 001=yes)
dect_freq_6	char(3)	Frequency status 6, swu, (000=no, 001=yes)
dect_freq_7	char(3)	Frequency status 7, swu, (000=no, 001=yes)
dect_freq_8	char(3)	Frequency status 8, swu, (000=no, 001=yes)

dect_freq_9	char(3)	Frequency status 9, swu, (000=no, 001=yes)
dect_freq_10	char(3)	Frequency status 10, swu, (000=no, 001=yes)
delay_comp	char(5)	Compensation, swu
hw_version	char(2)	HW version, swu
info	char(600)	Remarks
hopping_mode	char(3)	HOPPING_MODE, swu, coded (see hopping mode)
sw_version	char(13)	Software version, swu
fw_version	char(10)	Firmware Version, swu
broadband	char(3)	swu, coded
trans_power	char(3)	swu, coded
bs_type33	char(1)	swu
location	char(31)	location
base_station_type	char(3)	base_station_type
bs_port_on	char(3)	bs_port_on
bs_port_off	char(3)	bs_port_off
prefix_num	char(10)	
st_progress	char(3)	coded

10.2.1.19 CMI_BOARD

Table 42: CMI_BOARD

CMI_BOARD		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name, key
card_no	char(3)	SLMC No., swu
card_type	char(3)	SLMC Type, swu, coded (see CMI board)
slot	char(9)	pen, swu, key

system_name	char(16)	name, swu
system_id	char(10)	System no., swu
voice_announ_no	char(22)	Voice announcement number, swu
announ_timer	char(3)	Announcement timer, swu
store_calls	char(3)	Store calls, swu, coded (see CMI Values)
button_loc	char(3)	Key layout, swu, coded (see CMI Values)
encryption	char(3)	Code, swu, coded (see CMI Values)
comm_time	char(3)	Communication time, swu
roam_node	char(3)	Roaming-1 node, swu
roam_group	char(3)	Roaming-1 group, swu
deregist_no	char(12)	Logoff number, swu
regist_no	char(12)	Logon number, swu
connect_no	char(12)	SLMC Call Number, swu
version	char(3)	version, swu, coded (see Values for version in cmi_board)
language	char(3)	Language, swu, coded (see General Coded Values (0-10))
dpln_block1	char(4)	No. block 1, swu
dpln_block2	char(4)	No. block 2, swu
do_not_disturb	char(3)	do_not_disturb, swu, coded (000=no, 001=yes)
qadpcm	char(3)	QADPCM, swu
qdcl_signalling	char(3)	Fast communication, swu, coded (000=no, 001=yes)
pen_cnt	char(3)	Number of ports, swu
acl_timer	char(3)	ACL Verbindungszeit, swu
regist_release	char(3)	Enable login, swu, coded
pmid_handling	char(1)	PMID Handling, swu
early_roaming	char(3)	Early_Roaming, swu, coded
post_mort_mess	char(3)	Post_Mort_Mess, swu, coded
data_feature	char(3)	Data_Feature, swu, coded
ic_data_conn	char(2)	IC_Data_Conn, swu

lw_id	char(2)	Loadware-ID, swu
load_errmsg_dis	char(3)	swu, coded
max_card_no	char(3)	swu
mwi_signalling	char(3)	swu, coded
esc_substitution	char(3)	swu, coded
guest_board	char(3)	swu, coded (000=no, 001=yes)
switch_panel_1	char(3)	swu, coded
switch_panel_2	char(3)	swu, coded
switch_panel_3	char(3)	swu, coded
switch_panel_4	char(3)	swu, coded
switch_panel_5	char(3)	swu, coded
switch_panel_6	char(3)	swu, coded
switch_panel_7	char(3)	swu, coded
switch_panel_8	char(3)	swu, coded
switch_panel_9	char(3)	swu, coded
switch_panel_10	char(3)	swu, coded
switch_panel_11	char(3)	swu, coded
switch_panel_12	char(3)	swu, coded
switch_panel_13	char(3)	swu, coded
switch_panel_14	char(3)	swu, coded
switch_panel_15	char(3)	swu, coded
switch_panel_16	char(3)	swu, coded
switch_panel_17	char(3)	swu, coded
switch_panel_18	char(3)	swu, coded
switch_panel_19	char(3)	swu, coded
switch_panel_20	char(3)	swu, coded
switch_panel_21	char(3)	swu, coded
switch_panel_22	char(3)	swu, coded
switch_panel_23	char(3)	swu, coded
switch_panel_24	char(3)	swu, coded
max_subsc_pro_slc	char(5)	max_subsc_pro_slc
prefix_num	char(10)	

base_station_type	char(3)	coded

10.2.1.20 CMI_BOARD_SUB

Table 43: CMI_BOARD_SUB

CMI_BOARD_SUB		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name, key
slot	char(9)	pen, swu, key
ind	char(1)	number, swu, key
emc_no	char(5)	Code, swu
emc_name	char(16)	name, swu

10.2.1.21 CMI_QUER

Table 44: CMI_QUER

CMI_QUER		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name
domain	char(8)	domain
slot	char(9)	pen, swu
card_no	char(3)	SLMC No., swu
quer	char(12)	tienum, swu
dest_switch	char(4)	Dest. system, swu
system_id	char(10)	System no., swu

10.2.1.22 CMI_SYSTEM

Table 45: CMI_SYSTEM

CMI_SYSTEM

:EXPORT:SELECT		
unique_key	int	unique_key
system_name	char(16)	name, key
system_id	char(10)	System no.
voice_announ_no	char(22)	Voice announcement number
announ_timer	char(3)	Announcement timer
store_calls	char(3)	Store calls, coded (see CMI Values, Displayanzeige)
button_loc	char(3)	Key layout, coded (see CMI Values, Displayanzeige)
encryption	char(3)	Code, coded (see CMI Values, Displayanzeige)
comm_time	char(3)	Communication time
language	char(3)	Language, coded (see General Coded Values (0-10))
do_not_disturb	char(3)	do_not_disturb, coded (000=no, 001=yes)
acl_timer	char(3)	ACL Verbindungszeit
mwi_signalling	char(3)	coded
esc_substitution	char(3)	coded
max_subsc_no	char(5)	

10.2.1.23 CMI_SYSTEM_SUB

Table 46: CMI_SYSTEM_SUB

CMI_SYSTEM_SUB		
:EXPORT:SELECT		
unique_key	int	unique_key
system_name	char(16)	name, key
ind	char(1)	number, key
emc_no	char(5)	Code
emc_name	char(16)	name

10.2.1.24 CODED_VAL

Table 47: CODED_VAL

CODED_VAL		
:EXPORT:SELECT		
cod_val	char(3)	coded value
lang_id	char(5)	language id
lang_txt	char(150)	coded value text

10.2.1.25 COEX

Table 48: COEX

COEX		extern COS
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain, key
switch_name	char(4)	switch_name, key
coex	char(2)	external cos, swu, key
cosnum	char(4)	cos, swu
info	char(300)	info

10.2.1.26 COMPANY_CODE

Table 49: COMPANY_CODE

COMPANY_CODE		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
company	char(4)	company, swu
comp_ptr	char(2)	pointer to company code, swu
itr_group	char(2)	internal traffic restriction group, swu, key

10.2.1.27 COMPORG

Table 50: COMPORG

ODF Name: **COMPORG** Export: **Select** Import: **-** Operations:		
Field	Data type	Description
identifier	int	unique key
country	char(2)	country, where the company resides
company	char(30)	name of the company
address1	char(30)	address of the company line 1
address2	char(30)	address of the company line 2
org1	char(40)	name of the department, part 1
org2	char(40)	name of the department, part 2
org3	char(40)	name of the department, part 3
org4	char(40)	
org5	char(40)	
org6	char(40)	

10.2.1.28 COS

Table 51: COS

COS		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
cosnum	char(4)	cosnum, swu, key
switch_name	char(4)	switch_name, key
info	char(300)	info
c_mark_v01	char(3)	RTA1, swu, coded (000=no, 001=yes)
c_mark_v02	char(3)	RTA2, swu, coded (000=no, 001=yes)
c_mark_v03	char(3)	RTA3, swu, coded (000=no, 001=yes)

c_mark_v04	char(3)	RTA4, swu, coded (000=no, 001=yes)
c_mark_v05	char(3)	RTA5, swu, coded (000=no, 001=yes)
c_mark_v06	char(3)	RTA6, swu, coded (000=no, 001=yes)
c_mark_v07	char(3)	TA, swu, coded (000=no, 001=yes)
c_mark_v08	char(3)	RTA, swu, coded (000=no, 001=yes)
c_mark_v09	char(3)	NOCO, swu, coded (000=no, 001=yes)
c_mark_v10	char(3)	TTCR1, swu, coded (000=no, 001=yes)
c_mark_v11	char(3)	TTCR2, swu, coded (000=no, 001=yes)
c_mark_v12	char(3)	TTCR3, swu, coded (000=no, 001=yes)
c_mark_v13	char(3)	TTCR4, swu, coded (000=no, 001=yes)
c_mark_v14	char(3)	TTCR5, swu, coded (000=no, 001=yes)
c_mark_v15	char(3)	TTCR6, swu, coded (000=no, 001=yes)
c_mark_v16	char(3)	TNOTCR, swu, coded (000=no, 001=yes)
c_mark_v17	char(3)	RTIEA, swu, coded (000=no, 001=yes)
c_mark_v18	char(3)	NOTIE, swu, coded (000=no, 001=yes)
c_mark_v19	char(3)	TSUID, swu, coded (000=no, 001=yes)
c_mark_v20	char(3)	NOCHRG, swu, coded (000=no, 001=yes)
c_mark_v21	char(3)	PRCO, swu, coded (000=no, 001=yes)
c_mark_v22	char(3)	PRCO1, swu, coded (000=no, 001=yes)
c_mark_v23	char(3)	COIN, swu, coded (000=no, 001=yes)

c_mark_v24	char(3)	RKNOVR, swu, coded (000=no, 001=yes)
c_mark_v25	char(3)	RKOABS, swu, coded (000=no, 001=yes)
c_mark_v26	char(3)	KN, swu, coded (000=no, 001=yes)
c_mark_v27	char(3)	KNOVR, swu, coded (000=no, 001=yes)
c_mark_v28	char(3)	MB, swu, coded (000=no, 001=yes)
c_mark_v29	char(3)	COSXCD, swu, coded (000=no, 001=yes)
c_mark_v30	char(3)	Data, swu, coded (000=no, 001=yes)
c_mark_v31	char(3)	DICT, swu, coded (000=no, 001=yes)
c_mark_v32	char(3)	VC, swu, coded (000=no, 001=yes)
c_mark_v33	char(3)	DNDOVR, swu, coded (000=no, 001=yes)
c_mark_v34	char(3)	DIDBLK, swu, coded (000=no, 001=yes)
c_mark_v35	char(3)	TRACE, swu, coded (000=no, 001=yes)
c_mark_v36	char(3)	HOT, swu, coded (000=no, 001=yes)
c_mark_v37	char(3)	AHOT, swu, coded (000=no, 001=yes)
c_mark_v38	char(3)	PRISTN, swu, coded (000=no, 001=yes)
c_mark_v39	char(3)	NOANSA, swu, coded (000=no, 001=yes)
c_mark_v40	char(3)	OFHKRC, swu, coded (000=no, 001=yes)
c_mark_v41	char(3)	SPKR, swu, coded (000=no, 001=yes)
c_mark_v42	char(3)	vce, swu, coded (000=no, 001=yes)
c_mark_v43	char(3)	RSVLN, swu, coded (000=no, 001=yes)

c_mark_v44	char(3)	FWDNWK, swu, coded (000=no, 001=yes)
c_mark_v45	char(3)	ATRACE, swu, coded (000=no, 001=yes)
c_mark_v46	char(3)	MTRACE, swu, coded (000=no, 001=yes)
c_mark_v47	char(3)	RERING, swu, coded (000=no, 001=yes)
c_mark_v48	char(3)	AUTOV, swu, coded (000=no, 001=yes)
c_mark_v49	char(3)	RELCON, swu, coded (000=no, 001=yes)
c_mark_v50	char(3)	EOVR, swu, coded (000=no, 001=yes)
c_mark_v51	char(3)	REOVR, swu, coded (000=no, 001=yes)
c_mark_v52	char(3)	TRANSG, swu, coded (000=no, 001=yes)
c_mark_v53	char(3)	NOARNO, swu, coded (000=no, 001=yes)
c_mark_v54	char(3)	NOPTCD, swu, coded (000=no, 001=yes)
c_mark_v55	char(3)	LOCC, swu, coded (000=no, 001=yes)
c_mark_v56	char(3)	NATC, swu, coded (000=no, 001=yes)
c_mark_v57	char(3)	INATC, swu, coded (000=no, 001=yes)
c_mark_v58	char(3)	LSVC, swu, coded (000=no, 001=yes)
c_mark_v59	char(3)	NSVC, swu, coded (000=no, 001=yes)
c_mark_v60	char(3)	TOLLC, swu, coded (000=no, 001=yes)
c_mark_v61	char(3)	LOCCNV, swu, coded (000=no, 001=yes)
c_mark_v62	char(3)	TOLLOP, swu, coded (000=no, 001=yes)
c_mark_v63	char(3)	TESTEQ, swu, coded (000=no, 001=yes)

c_mark_v64	char(3)	OVCAT, swu, coded (000=no, 001=yes)
c_mark_v65	char(3)	SPECAR, swu, coded (000=no, 001=yes)
c_mark_v66	char(3)	STNBCAL, swu, coded (000=no, 001=yes)
c_mark_v67	char(3)	STNBDB, swu, coded (000=no, 001=yes)
c_mark_v68	char(3)	STNBMSN, swu, coded (000=no, 001=yes)
c_mark_v69	char(3)	msn, swu, coded (000=no, 001=yes)
c_mark_v70	char(3)	FWDBSY, swu, coded (000=no, 001=yes)
c_mark_v71	char(3)	CDRATN, swu, coded (000=no, 001=yes)
c_mark_v72	char(3)	CDRS, swu, coded (000=no, 001=yes)
c_mark_v73	char(3)	CDRC, swu, coded (000=no, 001=yes)
c_mark_v74	char(3)	CDRSTN, swu, coded (000=no, 001=yes)
c_mark_v75	char(3)	CDRIND, swu, coded (000=no, 001=yes)
c_mark_v76	char(3)	DRKEY, swu, coded (000=no, 001=yes)
c_mark_v77	char(3)	MULTRA, swu, coded (000=no, 001=yes)
c_mark_v78	char(3)	CDRINT, swu, coded (000=no, 001=yes)
c_mark_v79	char(3)	REVCHA, swu, coded (000=no, 001=yes)
c_mark_v80	char(3)	LOCCPV, swu, coded (000=no, 001=yes)
c_mark_v81	char(3)	TOLLPV, swu, coded (000=no, 001=yes)
c_mark_v82	char(3)	INATCP, swu, coded (000=no, 001=yes)
c_mark_v83	char(3)	MANOV, swu, coded (000=no, 001=yes)

c_mark_v84	char(3)	ANSYN, swu, coded (000=no, 001=yes)
c_mark_v85	char(3)	FWDFAS, swu, coded (000=no, 001=yes)
c_mark_v86	char(3)	FWSBAS, swu, coded (000=no, 001=yes)
c_mark_v87	char(3)	FWDDIR, swu, coded (000=no, 001=yes)
c_mark_v88	char(3)	KIESIG, swu, coded (000=no, 001=yes)
c_mark_v89	char(3)	ANITC, swu, coded (000=no, 001=yes)
c_mark_v90	char(3)	ANIPS, swu, coded (000=no, 001=yes)
c_mark_v91	char(3)	ANICT, swu, coded (000=no, 001=yes)
c_mark_v92	char(3)	ANISS, swu, coded (000=no, 001=yes)
c_mark_v93	char(3)	CFSWS, swu, coded (000=no, 001=yes)
c_mark_v94	char(3)	FWDECA, swu, coded (000=no, 001=yes)
c_mark_v95	char(3)	DEC, swu, coded (000=no, 001=yes)
c_mark_v96	char(3)	RECOIG, swu, coded (000=no, 001=yes)
c_mark_v97	char(3)	SUBEPP, swu, coded (000=no, 001=yes)
c_mark_v98	char(3)	MFPAIC, swu, coded (000=no, 001=yes)
c_mark_v99	char(3)	MFPIAC, swu, coded (000=no, 001=yes)
c_mark_v100	char(3)	MFPSIC, swu, coded (000=no, 001=yes)
c_mark_v101	char(3)	MFPSTC, swu, coded (000=no, 001=yes)
c_mark_v102	char(3)	FWDDEXT, swu, coded (000=no, 001=yes)
c_mark_v103	char(3)	CCBS, swu, coded (000=no, 001=yes)

c_mark_v104	char(3)	COLR, swu, coded (000=no, 001=yes)
c_mark_v105	char(3)	CW, swu, coded (000=no, 001=yes)
c_mark_v106	char(3)	OETEL, swu, coded (000=no, 001=yes)
c_mark_v107	char(3)	SCRNCA, swu, coded (000=no, 001=yes)
c_mark_v108	char(3)	SCRNCB, swu, coded (000=no, 001=yes)
c_mark_v109	char(3)	SCRSPC, swu, coded (000=no, 001=yes)
c_mark_v110	char(3)	MTEQ, swu, coded (000=no, 001=yes)
c_mark_v111	char(3)	FWDBAR, swu, coded (000=no, 001=yes)
c_mark_v112	char(3)	GRPCAL, swu, coded (000=no, 001=yes)
c_mark_v113	char(3)	BDCP, swu, coded (000=no, 001=yes)
c_mark_v114	char(3)	MCIDM, swu, coded (000=no, 001=yes)
c_mark_v115	char(3)	MCIDA, swu, coded (000=no, 001=yes)
c_mark_v116	char(3)	TP, swu, coded (000=no, 001=yes)
c_mark_v117	char(3)	UUS1IM, swu, coded (000=no, 001=yes)
c_mark_v118	char(3)	UUS3, swu, coded (000=no, 001=yes)
c_mark_v119	char(3)	CTLS, swu, coded (000=no, 001=yes)
c_mark_v120	char(3)	SUTVA, swu, coded (000=no, 001=yes)
c_mark_d01	char(3)	RTA1, swu, coded (000=no, 001=yes)
c_mark_d02	char(3)	RTA2, swu, coded (000=no, 001=yes)
c_mark_d03	char(3)	RTA3, swu, coded (000=no, 001=yes)

c_mark_d04	char(3)	RTA4, swu, coded (000=no, 001=yes)
c_mark_d05	char(3)	RTA5, swu, coded (000=no, 001=yes)
c_mark_d06	char(3)	RTA6, swu, coded (000=no, 001=yes)
c_mark_d07	char(3)	TA, swu, coded (000=no, 001=yes)
c_mark_d08	char(3)	RTA, swu, coded (000=no, 001=yes)
c_mark_d09	char(3)	NOCO, swu, coded (000=no, 001=yes)
c_mark_d10	char(3)	TTCR1, swu, coded (000=no, 001=yes)
c_mark_d11	char(3)	TTCR2, swu, coded (000=no, 001=yes)
c_mark_d12	char(3)	TTCR3, swu, coded (000=no, 001=yes)
c_mark_d13	char(3)	TTCR4, swu, coded (000=no, 001=yes)
c_mark_d14	char(3)	TTCR5, swu, coded (000=no, 001=yes)
c_mark_d15	char(3)	TTCR6, swu, coded (000=no, 001=yes)
c_mark_d16	char(3)	TNOTCR, swu, coded (000=no, 001=yes)
c_mark_d17	char(3)	RTIEA, swu, coded (000=no, 001=yes)
c_mark_d18	char(3)	NOTIE, swu, coded (000=no, 001=yes)
c_mark_d19	char(3)	TSUID, swu, coded (000=no, 001=yes)
c_mark_d20	char(3)	NOCHRG, swu, coded (000=no, 001=yes)
c_mark_d21	char(3)	DIDBLK, swu, coded (000=no, 001=yes)
c_mark_d22	char(3)	HOT, swu, coded (000=no, 001=yes)
c_mark_d23	char(3)	NOANSA, swu, coded (000=no, 001=yes)

c_mark_d24	char(3)	BASIC, swu, coded (000=no, 001=yes)
c_mark_d25	char(3)	INTWOR, swu, coded (000=no, 001=yes)
c_mark_d26	char(3)	DSM, swu, coded (000=no, 001=yes)
c_mark_d27	char(3)	OVCAT, swu, coded (000=no, 001=yes)
c_mark_d28	char(3)	SPECAR, swu, coded (000=no, 001=yes)
c_mark_d29	char(3)	STNBCAL, swu, coded (000=no, 001=yes)
c_mark_d30	char(3)	STNBDB, swu, coded (000=no, 001=yes)
c_mark_d31	char(3)	STNBMSN, swu, coded (000=no, 001=yes)
c_mark_d32	char(3)	msn, swu, coded (000=no, 001=yes)
c_mark_d33	char(3)	CDRSTN, swu, coded (000=no, 001=yes)
c_mark_d34	char(3)	MULTRA, swu, coded (000=no, 001=yes)
c_mark_d35	char(3)	CDRINT, swu, coded (000=no, 001=yes)
c_mark_d36	char(3)	FWDFAS, swu, coded (000=no, 001=yes)
c_mark_d37	char(3)	FWSBAS, swu, coded (000=no, 001=yes)
c_mark_d38	char(3)	FWDDIR, swu, coded (000=no, 001=yes)
c_mark_d39	char(3)	KIESIG, swu, coded (000=no, 001=yes)
c_mark_d40	char(3)	DEC, swu, coded (000=no, 001=yes)
c_mark_d41	char(3)	INATCP, swu, coded (000=no, 001=yes)
c_mark_d42	char(3)	LOCCPV, swu, coded (000=no, 001=yes)
c_mark_d43	char(3)	TOLLPV, swu, coded (000=no, 001=yes)

c_mark_d44	char(3)	CDRIND, swu, coded (000=no, 001=yes)
c_mark_d45	char(3)	CCBS, swu, coded (000=no, 001=yes)
c_mark_d46	char(3)	COLR, swu, coded (000=no, 001=yes)
c_mark_d47	char(3)	MCIDM, swu, coded (000=no, 001=yes)
c_mark_d48	char(3)	MCIDA, swu, coded (000=no, 001=yes)
c_mark_d49	char(3)	UUS1IM, swu, coded (000=no, 001=yes)
c_mark_d50	char(3)	UUS3, swu, coded (000=no, 001=yes)
c_mark_d51	char(3)	c_mark_d51, swu, coded (000=no, 001=yes)
c_mark_f01	char(3)	RTA1, swu, coded (000=no, 001=yes)
c_mark_f02	char(3)	RTA2, swu, coded (000=no, 001=yes)
c_mark_f03	char(3)	RTA3, swu, coded (000=no, 001=yes)
c_mark_f04	char(3)	RTA4, swu, coded (000=no, 001=yes)
c_mark_f05	char(3)	RTA5, swu, coded (000=no, 001=yes)
c_mark_f06	char(3)	RTA6, swu, coded (000=no, 001=yes)
c_mark_f07	char(3)	TA, swu, coded (000=no, 001=yes)
c_mark_f08	char(3)	RTA, swu, coded (000=no, 001=yes)
c_mark_f09	char(3)	NOCO, swu, coded (000=no, 001=yes)
c_mark_f10	char(3)	TTCR1, swu, coded (000=no, 001=yes)
c_mark_f11	char(3)	TTCR2, swu, coded (000=no, 001=yes)
c_mark_f12	char(3)	TTCR3, swu, coded (000=no, 001=yes)

c_mark_f13	char(3)	TTCR4, swu, coded (000=no, 001=yes)
c_mark_f14	char(3)	TTCR5, swu, coded (000=no, 001=yes)
c_mark_f15	char(3)	TTCR6, swu, coded (000=no, 001=yes)
c_mark_f16	char(3)	TNOTCR, swu, coded (000=no, 001=yes)
c_mark_f17	char(3)	RTIEA, swu, coded (000=no, 001=yes)
c_mark_f18	char(3)	NOTIE, swu, coded (000=no, 001=yes)
c_mark_f19	char(3)	TSUID, swu, coded (000=no, 001=yes)
c_mark_f20	char(3)	NOCHRG, swu, coded (000=no, 001=yes)
c_mark_f21	char(3)	DIDBLK, swu, coded (000=no, 001=yes)
c_mark_f22	char(3)	NOANSA, swu, coded (000=no, 001=yes)
c_mark_f23	char(3)	fax, swu, coded (000=no, 001=yes)
c_mark_f24	char(3)	BASIC, swu, coded (000=no, 001=yes)
c_mark_f25	char(3)	INTWOR, swu, coded (000=no, 001=yes)
c_mark_f26	char(3)	OFTFS, swu, coded (000=no, 001=yes)
c_mark_f27	char(3)	OVCAT, swu, coded (000=no, 001=yes)
c_mark_f28	char(3)	SPECAR, swu, coded (000=no, 001=yes)
c_mark_f29	char(3)	STNBCAL, swu, coded (000=no, 001=yes)
c_mark_f30	char(3)	STNBDB, swu, coded (000=no, 001=yes)
c_mark_f31	char(3)	STNBMSN, swu, coded (000=no, 001=yes)
c_mark_f32	char(3)	msn, swu, coded (000=no, 001=yes)

c_mark_f33	char(3)	CDRSTN, swu, coded (000=no, 001=yes)
c_mark_f34	char(3)	MULTRA, swu, coded (000=no, 001=yes)
c_mark_f35	char(3)	CDRINT, swu, coded (000=no, 001=yes)
c_mark_f36	char(3)	FWDFAS, swu, coded (000=no, 001=yes)
c_mark_f37	char(3)	FWSBAS, swu, coded (000=no, 001=yes)
c_mark_f38	char(3)	FWDDIR, swu, coded (000=no, 001=yes)
c_mark_f39	char(3)	KIESIG, swu, coded (000=no, 001=yes)
c_mark_f40	char(3)	DEC, swu, coded (000=no, 001=yes)
c_mark_f41	char(3)	INATCP, swu, coded (000=no, 001=yes)
c_mark_f42	char(3)	LOCCPV, swu, coded (000=no, 001=yes)
c_mark_f43	char(3)	TOLLPV, swu, coded (000=no, 001=yes)
c_mark_f44	char(3)	CDRIND, swu, coded (000=no, 001=yes)
c_mark_f45	char(3)	CCBS, swu, coded (000=no, 001=yes)
c_mark_f46	char(3)	COLR, swu, coded (000=no, 001=yes)
c_mark_f47	char(3)	MCIDM, swu, coded (000=no, 001=yes)
c_mark_f48	char(3)	MCIDA, swu, coded (000=no, 001=yes)
c_mark_f49	char(3)	UUS1IM, swu, coded (000=no, 001=yes)
c_mark_f50	char(3)	UUS3, swu, coded (000=no, 001=yes)
c_mark_f51	char(3)	c_mark_f51, swu, coded (000=no, 001=yes)
info2	char(50)	AMO Description, swu

10.2.1.29 COSXT

Table 52: COSXT

COSXT		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name, key
berumgrpnum	char(2)	berumgrpnum, swu, key
info	char(300)	info
tim_cos1_mon	char(5)	Monday Cos1, swu
tim_cos1_mon_act	char(3)	Monday Cos 1 activated, swu, coded (000=no, 001=yes)
tim_cos2_mon	char(5)	Monday Cos2, swu
tim_cos2_mon_act	char(3)	Monday Cos 2 activated, swu, coded (000=no, 001=yes)
tim_cos1_tue	char(5)	Tuesday Cos 1, swu
tim_cos1_tue_act	char(3)	Tuesday Cos 1 activated, swu, coded (000=no, 001=yes)
tim_cos2_tue	char(5)	Tuesday Cos 2, swu
tim_cos2_tue_act	char(3)	Tuesday Cos 2 activated, swu, coded (000=no, 001=yes)
tim_cos1_wed	char(5)	Wednesday Cos 1, swu
tim_cos1_wed_act	char(3)	Wednesday Cos 1 activated, swu, coded (000=no, 001=yes)
tim_cos2_wed	char(5)	Wednesday Cos 2, swu
tim_cos2_wed_act	char(3)	Wednesday Cos 2 activated, swu, coded (000=no, 001=yes)
tim_cos1_thu	char(5)	Thursday Cos 1, swu
tim_cos1_thu_act	char(3)	Thursday Cos 1 activated, swu, coded (000=no, 001=yes)

tim_cos2_thu	char(5)	Thursday Cos 2, swu
tim_cos2_thu_act	char(3)	Thursday Cos 2 activated, swu, coded (000=no, 001=yes)
tim_cos1_fri	char(5)	Friday Cos 1, swu
tim_cos1_fri_act	char(3)	Friday Cos 1 activated, swu, coded (000=no, 001=yes)
tim_cos2_fri	char(5)	Friday Cos 2, swu
tim_cos2_fri_act	char(3)	Friday Cos 2 activated, swu, coded (000=no, 001=yes)
tim_cos1_sat	char(5)	Saturday Cos 1, swu
tim_cos1_sat_act	char(3)	Saturday Cos 1 activated, swu, coded (000=no, 001=yes)
tim_cos2_sat	char(5)	Saturday Cos 2, swu
tim_cos2_sat_act	char(3)	Saturday Cos 2 activated, swu, coded (000=no, 001=yes)
tim_cos1_sun	char(5)	Sunday Cos 1, swu
tim_cos1_sun_act	char(3)	Sunday Cos 1 activated, swu, coded (000=no, 001=yes)
tim_cos2_sun	char(5)	Sunday Cos 2, swu
tim_cos2_sun_act	char(3)	Sunday Cos 2 activated, swu, coded (000=no, 001=yes)
sync_by	char(4)	sync by
sync_time	int	sync time

10.2.1.30 CTRYNUM

Table 53: CTRYNUM

CTRYNUM		
:EXPORT:SELECT		
:IMPORT:INSERT: DELETE:UPDATE		
unique_key	int	unique_key
src_country	char(2)	src_country, key

dest_country	char(2)	dest_country, key
id	char(5)	id
info	char(300)	info

10.2.1.31 DEVCOMB

Table 54: DEVCOMB

DEVCOMB		
:EXPORT:SELECT		
unique_key	int	unique_key, swu
exact_dev_typ	char(3)	exact device type, swu, coded (see *Devices 200-249, device types)
devtyp1	char(3)	devtyp1, swu, coded (see *Devices 200-249, device types)
idcard_rd	char(3)	idcard_rd, swu, coded (000=no, 001=yes)
addon_typ	char(3)	addon_typ, swu, coded (see Addon type)
handsfree	char(3)	handsfree, swu, coded (000=no, 001=yes)
sys_verify	int	sys_verify, swu
display	char(3)	display, swu, coded (000=no, 001=yes)
devtyp2	char(3)	devtyp2, swu, coded (see *Devices 200-249, device types)
devtyp3	char(3)	devtyp3, swu, coded (see *Devices 200-249, device types)
devtyp4	char(3)	devtyp4, swu, coded (see *Devices 200-249, device types)
devtyp5	char(3)	devtyp5, swu, coded (see *Devices 200-249, device types)
adaptor	char(3)	adaptor, swu, coded
funct_included	char(3)	funct_included, swu, coded
cnt_of_devices	int	cnt_of_devices, swu
devtyp6	char(3)	devtyp6, swu, coded (see *Devices 200-249, device types)

10.2.1.32 DEVCONST

Table 55: DEVCONST

DEVCONST		
:EXPORT:SELECT		
unique_key	int	unique_key
devconame	char(20)	Device combination, key
info	char(300)	info
chargeindic	char(2)	Basic charge class
exact_dev_typ	char(3)	device type, coded (see *Devices 200-249, device types)
exact_dev_text	char(30)	exact_dev_text
devtyp1	char(3)	Device class, coded (see *Devices 200-249, device types)
devtext1	char(30)	Device1
idcard_rd	char(3)	idcard_rd, coded (000=no, 001=yes)
addon_typ	char(3)	Add-on device, coded (see Addon type)
addon_text	char(30)	addon_text
handsfree	char(3)	handsfree, coded (000=no, 001=yes)
sys_verify	int	sys_verify
display	char(3)	display, coded (000=no, 001=yes)
devtyp2	char(3)	Data, coded (see *Devices 200-249, device types)
devtext2	char(30)	Device2
devtyp3	char(3)	fax, coded (see *Devices 200-249, device types)
devtext3	char(30)	Device3
devtyp4	char(3)	TTX, coded (see *Devices 200-249, device types)

devtext4	char(30)	Device4
devtyp5	char(3)	VTX, coded (see *Devices 200-249, device types)
devtext5	char(30)	Device5
adaptor_1	char(3)	Adapter 1, coded (see *Devices 200-249)
adaptor_2	char(3)	Adapter 2, coded (see *Devices 200-249)
consistent	char(3)	consistent, coded (000=no, 001=yes)
funct_included	char(3)	funct_included, coded (000=no, 001=yes)
cnt_of_devices	int	cnt_of_devices
opticom	char(3)	ComManager, coded (000=no, 001=yes)
headset	char(3)	Headset, coded (see headset,devconst)
recorder	char(3)	Headset Key, coded (see recorder,devconst)
devtyp6	char(3)	Other, coded (see *Devices 200-249, device types)
devtext6	char(30)	Device6

10.2.1.33 DIMSU

Table 56: DIMSU

DIMSU		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
dimsu_typ	char(3)	dimsu_typ, swu, key, coded (see DIMSU's, additional dimsu, Unity's dimsu)
max_dimsu_val	int	max_dimsu_val, swu
used_dimsu_val	int	used_dimsu_val, swu

vms_flag	char(1)	vms_flag, swu
origin	char(1)	origin, swu
actual_dimsu_val	int	actual value, swu
std_dimsu_val	int	standard value, swu
sysmax_dimsu_val	int	system specific max value, swu

10.2.1.34 DISPLAY_NAME

Table 57: DISPLAY_NAME

DISPLAY_NAME		
:EXPORT:SELECT		
:IMPORT:INSERT: DELETE:UPDATE		
unique_key	int	unique_key
switch_name	char(4)	(required) System, key
domain	char(8)	(required) domain
extension	char(12)	(required) Station No., swu, key
name	char(31)	(required) Display Name, swu
department	char(25)	Department, swu
position	int	Index, swu, key
charcon	char(3)	charcon, swu, coded(see display_name,charcon)
displayed	char(3)	coded (000, 001)
subdomain	char(8)	subdomain
template_id	int	unique key of the record in templates table
subscr_key	int	unique key of the record in persdat table

10.2.1.35 DOMAIN

Table 58: DOMAIN

DOMAIN

:EXPORT:SELECT	:IMPORT:	INSERT:DELETE:UPDATE
unique_key	int	unique_key
domain	char(8)	domain, key
info	char(300)	info
tienum	char(6)	tienum
epnp_support	char(3)	epnp_support, coded
dialplan_type1	char(3)	dialplan type1, coded
tenant_group	char(20)	Note: **Manager** only

10.2.1.36 DOWNLOAD_CLASS

Table 59: DOWNLOAD_CLASS

DOWNLOAD_CLASS		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
number	char(3)	number, swu, key
info	char(300)	info

10.2.1.37 DOWNLOAD_PARAMS

Table 60: DOWNLOAD_PARAMS

DOWNLOAD_PARAMS		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
number	char(3)	number, swu, key
info	char(300)	info

10.2.1.38 DOWNLOAD_SUPP

Table 61: DOWNLOAD_SUPP

DOWNLOAD_SUPP		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
ph28	char(2)	ph28, swu
ph29	char(2)	ph29, swu
ph2A	char(2)	ph2A, swu
ph2B	char(2)	ph2B, swu
ph2C	char(2)	ph2C, swu
ph2D	char(2)	ph2D, swu
ph2E	char(2)	ph2E, swu
ph2F	char(2)	ph2F, swu
ph30	char(2)	ph30, swu
ph31	char(2)	ph31, swu
ph32	char(2)	ph32, swu
ph33	char(2)	ph33, swu
ph34	char(2)	ph34, swu
ph35	char(2)	ph35, swu
ph36	char(2)	ph36, swu
ph37	char(2)	ph37, swu
ph38	char(2)	ph38, swu
ph39	char(2)	ph39, swu
ph3A	char(2)	ph3A, swu
ph3B	char(2)	ph3B, swu
ph3C	char(2)	ph3C, swu
ph3D	char(2)	ph3D, swu
ph3E	char(2)	ph3E, swu
ph3F	char(2)	ph3F, swu

ph40	char(2)	ph40, swu
ph41	char(2)	ph41, swu
ph42	char(2)	ph42, swu
ph43	char(2)	ph43, swu
ph44	char(2)	ph44, swu
ph45	char(2)	ph45, swu
ph46	char(2)	ph46, swu
ph47	char(2)	ph47, swu
ph48	char(2)	ph48, swu
ph49	char(2)	ph49, swu
ph4A	char(2)	ph4A, swu
ph4B	char(2)	ph4B, swu
ph4C	char(2)	ph4C, swu
ph4D	char(2)	ph4D, swu
ph4E	char(2)	ph4E, swu
ph4F	char(2)	ph4F, swu
ph50	char(2)	ph50, swu
ph51	char(2)	ph51, swu
ph52	char(2)	ph52, swu
ph53	char(2)	ph53, swu
ph54	char(2)	ph54, swu
ph55	char(2)	ph55, swu
ph56	char(2)	ph56, swu
ph57	char(2)	ph57, swu
ph58	char(2)	ph58, swu
ph59	char(2)	ph59, swu
ph5A	char(2)	ph5A, swu
ph5B	char(2)	ph5B, swu
ph5C	char(2)	ph5C, swu
ph5D	char(2)	ph5D, swu
ph5E	char(2)	ph5E, swu
ph5F	char(2)	ph5F, swu

ph60	char(2)	ph60, swu
ph61	char(2)	ph61, swu
ph62	char(2)	ph62, swu
ph63	char(2)	ph63, swu

10.2.1.39 DYNAMIC_KEYDAT

Table 62: DYNAMIC_KEYDAT

DYNAMIC_KEYDAT		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name, key
extension	char(12)	extension, swu, key
key_number	char(3)	number, swu, key
key_deviation	char(3)	key_deviation, swu, coded (000=no, 001=yes)
key_function	char(3)	key_function, swu, coded (see key functions (850-899), key functions (920-949))
key_destination	char(22)	key_destination, swu
key_protect	char(3)	Rejection, swu, coded (000=no, 001=yes)
ring_opt	char(3)	Ring Option, swu, coded
aics	char(3)	Automatic call signaling, swu, coded
key_direct	char(3)	Voice Calling, swu, coded (000=no, 001=yes)
dss_wo_overr	char(3)	swu,coded (000=no, 001=yes)
label_text	char(12)	label text, swu
label_option	char(3)	label option, swu, coded
dssline	char(3)	dssline, swu, coded
line_tone	char(3)	swu, coded

ring_type	char(2)	swu
ring_volume	char(2)	swu
busy_led	char(3)	swu, coded
same_prio	char(3)	swu, coded
prio_call	char(3)	swu, coded
blf	char(3)	swu, coded

10.2.1.40 E164_NUM

Table 63: E164_NUM

E164_NUM		
:EXPORT:SELECT		
switch_name	char(4)	switch_name, key
extension	char(6)	extension, swu, key
isdn_cc	char(4)	isdn_cc, swu
isdn_ac	char(6)	isdn_ac, swu
isdn_lc	char(10)	isdn_lc, swu
isdn_skip_digit	char(2)	isdn_skip_digits, swu
e164	char(18)	internally counted E164 number; cannot be used in WHERE clause of SQL select statement

10.2.1.41 ERRORS

Table 64: ERRORS

ERRORS		
:EXPORT:SELECT		
pabx_id	int	identification of pabx
l_serial	int	l_serial
msg_id	char(5)	error message number
msg_lauf	char(5)	error message sequence number
priority	char(5)	error message priority

action	char(8)	error message action
abs_mod	char(5)	server id
event	char(8)	primary event
subevent	char(22)	subevent
card_ref	char(30)	card references
bg_stand	char(2)	
fw_kenn	char(6)	
dat_tim	int	creation time of error message
dev_cl_hex	char(50)	device class hex string
dev_cl_gr	int	device group
proc_state	int	
due_date	int	
responsible_name	char(20)	
hts_flag	int	
server_time	int	
ltu	int	

10.2.1.42 FIRMDAT

Table 65: FIRMDAT

FIRMDAT		
:EXPORT:SELECT		
:IMPORT:INSERT: DELETE:UPDATE		
unique_key	int	unique_key
country	char(2)	(required) country, key
company	char(30)	(required) company, key
address1	char(30)	(required) Address 1
address2	char(30)	Address 2

10.2.1.43 HUNTGRP

Table 66: HUNTGRP

HUNTGRP		Huntgroup/Sammelanschluß
:EXPORT:SELECT		
group_idx	int	group index, key
domain	char(8)	domain, key
switch_name	char(4)	switch_name, key
huntgrpnum	char(4)	hunt group number
default_num	char(3)	default num (001=yes, 000=no)
cps_str	char(23)	cps string
dpln_str	char(16)	dpln string
type	char(3)	type (001=Master, 000=regular hunt group)
extension	char(12)	add. extension for regular huntgroup
position	int	position in huntgroup
service	char(3)	type of service (i90=voice, 212=fax, D13=Data
search_criteria	char(3)	type of searching (004=cyclic, 005=li
call_sel_cap	char(3)	call selection (001=yes, 000=no)
mailbox	char(6)	extension number of mailbox
callforward	char(3)	callforwarding (001=yes, 000=no)
varao	char(3)	var.call selector (001=yes, 000=no)
name	char(16)	name at the display
info	char(50)	Description

10.2.1.44 HUNTGRP_AC

Table 67: HUNTGRP_AC

HUNTGRP_AC	
:EXPORT:SELECT	
unique_key	int
switch_name	char(4)

domain	char(8)
group_idx	int
huntgrpnum	char(12)
default_num	char(3)
cps_str	char(23)
dpln_str	char(16)
type	char(3)

10.2.1.45 HUNTGRP_DPLN_CPS

Table 68: HUNTGRP_DPLN_CPS

HUNTGRP_DPLN_CPS	
:EXPORT:SELECT	
unique_key	int
switch_name	char(4)
domain	char(8)
group_idx	int
dpln0	char(3)
dpln1	char(3)
dpln2	char(3)
dpln3	char(3)
dpln4	char(3)
dpln5	char(3)
dpln6	char(3)
dpln7	char(3)
dpln8	char(3)
dpln9	char(3)
dpln10	char(3)
dpln11	char(3)
dpln12	char(3)
dpln13	char(3)
dpln14	char(3)

dpln15	char(3)
cps2	char(3)
cps3	char(3)
cps4	char(3)
cps5	char(3)
cps6	char(3)
cps7	char(3)
cps8	char(3)
cps9	char(3)
cps10	char(3)
cps11	char(3)
cps12	char(3)
cps22	char(3)

10.2.1.46 HUNTGRP_SERVICE

Table 69: HUNTGRP_SERVICE

HUNTGRP_SERVICE		
:EXPORT:SELECT		
:IMPORT:INSERT:DELETE:UPDATE		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
group_idx	int	group_idx, key
service	char(3)	
search_criteria	char(3)	
call_sel_cap	char(3)	
mailbox	char(6)	
callforward	char(3)	
varao	char(3)	
cohg	char(3)	
autodig	char(3)	

customer	char(3)
cfd_net	char(3)
overflow_type	char(3)
overflow_ext	char(22)
overflow_loop	char(3)
disptype	char(3)
leave_grp	char(3)
lstmboff	char(3)
itr	char(2)
busy_cond	char(3)
check_cfu_avail	char(3)

10.2.1.47 HUNTGRP_SUB

Table 70: HUNTGRP_SUB

HUNTGRP_SUB		
:EXPORT:SELECT		
unique_key	int	unique_key
huntgrpnum	char(12)	huntgrpnum, swu
switch_name	char(4)	switch_name, key
domain	char(8)	domain
group_idx	int	group_idx, key
service	char(3)	service, swu, key, coded (see cfw, service)
extension	char(22)	member extension, swu, key
position	int	position, swu
ref_mem	char(3)	call forwarding, swu, coded (000=no, 001=yes)
remote_switch	char(4)	Remote System, swu
remote_domain	char(8)	Remote Domain, swu
template_id	int	unique key of the record in templates table

10.2.1.48 KNPRE

Table 71: KNPRE

KNPRE		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
dpln	char(2)	
npi	char(3)	
ton	char(3)	
exit	char(6)	
prefix	char(4)	
info	char(300)	

10.2.1.49 KZNR

Table 72: KZNR

KZNR		speed dial / Kurzrufnummern
SELECT		
INSERT		
DELETE		
UPDATE		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
num	char(2)	num, swu, key
info	char(300)	info
kznr_index	char(3)	kznr_index, swu, key
long_num	char(22)	extension, swu
num_digit_supres	char(2)	num_digit_supres, swu
domain	char(8)	domain

10.2.1.50 KZNR_LIST

Table 73: KZNR_LIST

KZNR_LIST		speed dial list / Kurzrufnummernliste
SELECT		
INSERT		
DELETE		
UPDATE		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name, key
num	char(2)	num, swu, key
first_idx	char(3)	first_idx, swu
last_idx	char(3)	last_idx, swu
info	char(300)	info
sync_by	char(4)	synchronised by
sync_time	int	sync_time

10.2.1.51 LBER

Table 74: LBER

LBER		
:EXPORT:SELECT		
unique_key	int	unique_key
lber	char(2)	LCR authorisation, swu, key
info	char(300)	info

10.2.1.52 LCOS_DATA

Table 75: LCOS_DATA

LCOS_DATA		
:EXPORT:SELECT		
unique_key	int	unique_key

switch_name	char(4)	switch_name, key
domain	char(8)	domain
lcos	char(4)	LCOS data, swu, key
info	char(300)	info
l1	char(3)	LAUTH1, swu, coded (000=no, 001=yes)
l2	char(3)	LAUTH2, swu, coded (000=no, 001=yes)
l3	char(3)	LAUTH3, swu, coded (000=no, 001=yes)
l4	char(3)	LAUTH4, swu, coded (000=no, 001=yes)
l5	char(3)	LAUTH5, swu, coded (000=no, 001=yes)
l6	char(3)	LAUTH6, swu, coded (000=no, 001=yes)
l7	char(3)	LAUTH7, swu, coded (000=no, 001=yes)
l8	char(3)	LAUTH8, swu, coded (000=no, 001=yes)
l9	char(3)	LAUTH9, swu, coded (000=no, 001=yes)
l10	char(3)	LAUTH10, swu, coded (000=no, 001=yes)
l11	char(3)	LAUTH11, swu, coded (000=no, 001=yes)
l12	char(3)	LAUTH12, swu, coded (000=no, 001=yes)
l13	char(3)	LAUTH13, swu, coded (000=no, 001=yes)
l14	char(3)	LAUTH14, swu, coded (000=no, 001=yes)
l15	char(3)	LAUTH15, swu, coded (000=no, 001=yes)
l16	char(3)	LAUTH16, swu, coded (000=no, 001=yes)
l17	char(3)	LAUTH17, swu, coded (000=no, 001=yes)
l18	char(3)	LAUTH18, swu, coded (000=no, 001=yes)
l19	char(3)	LAUTH19, swu, coded (000=no, 001=yes)
l20	char(3)	LAUTH20, swu, coded (000=no, 001=yes)
l21	char(3)	LAUTH21, swu, coded (000=no, 001=yes)
l22	char(3)	LAUTH22, swu, coded (000=no, 001=yes)
l23	char(3)	LAUTH23, swu, coded (000=no, 001=yes)
l24	char(3)	LAUTH24, swu, coded (000=no, 001=yes)
l25	char(3)	LAUTH25, swu, coded (000=no, 001=yes)
l26	char(3)	LAUTH26, swu, coded (000=no, 001=yes)
l27	char(3)	LAUTH27, swu, coded (000=no, 001=yes)
l28	char(3)	LAUTH28, swu, coded (000=no, 001=yes)

l29	char(3)	LAUTH29, swu, coded (000=no, 001=yes)
l30	char(3)	LAUTH30, swu, coded (000=no, 001=yes)
l31	char(3)	LAUTH31, swu, coded (000=no, 001=yes)
l32	char(3)	LAUTH32, swu, coded (000=no, 001=yes)
l33	char(3)	LAUTH33, swu, coded (000=no, 001=yes)
l34	char(3)	LAUTH34, swu, coded (000=no, 001=yes)
l35	char(3)	LAUTH35, swu, coded (000=no, 001=yes)
l36	char(3)	LAUTH36, swu, coded (000=no, 001=yes)
l37	char(3)	LAUTH37, swu, coded (000=no, 001=yes)
l38	char(3)	LAUTH38, swu, coded (000=no, 001=yes)
l39	char(3)	LAUTH39, swu, coded (000=no, 001=yes)
l40	char(3)	LAUTH40, swu, coded (000=no, 001=yes)
l41	char(3)	LAUTH41, swu, coded (000=no, 001=yes)
l42	char(3)	LAUTH42, swu, coded (000=no, 001=yes)
l43	char(3)	LAUTH43, swu, coded (000=no, 001=yes)
l44	char(3)	LAUTH44, swu, coded (000=no, 001=yes)
l45	char(3)	LAUTH45, swu, coded (000=no, 001=yes)
l46	char(3)	LAUTH46, swu, coded (000=no, 001=yes)
l47	char(3)	LAUTH47, swu, coded (000=no, 001=yes)
l48	char(3)	LAUTH48, swu, coded (000=no, 001=yes)
l49	char(3)	LAUTH49, swu, coded (000=no, 001=yes)
l50	char(3)	LAUTH50, swu, coded (000=no, 001=yes)
l51	char(3)	LAUTH51, swu, coded (000=no, 001=yes)
l52	char(3)	LAUTH52, swu, coded (000=no, 001=yes)
l53	char(3)	LAUTH53, swu, coded (000=no, 001=yes)
l54	char(3)	LAUTH54, swu, coded (000=no, 001=yes)
l55	char(3)	LAUTH55, swu, coded (000=no, 001=yes)
l56	char(3)	LAUTH56, swu, coded (000=no, 001=yes)
l57	char(3)	LAUTH57, swu, coded (000=no, 001=yes)
l58	char(3)	LAUTH58, swu, coded (000=no, 001=yes)
l59	char(3)	LAUTH59, swu, coded (000=no, 001=yes)
l60	char(3)	LAUTH60, swu, coded (000=no, 001=yes)

l61	char(3)	LAUTH61, swu, coded (000=no, 001=yes)
l62	char(3)	LAUTH62, swu, coded (000=no, 001=yes)
l63	char(3)	LAUTH63, swu, coded (000=no, 001=yes)
l64	char(3)	LAUTH64, swu, coded (000=no, 001=yes)
sync_by	char(4)	copied from
sync_time	int	on
info2	char(50)	AMO Description, swu

10.2.1.53 LCOS_VOICE

Table 76: LCOS_VOICE

LCOS_VOICE		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
lcos	char(4)	LCOS voice, swu, key
info	char(300)	info
copin	char(2)	copin, swu
l1	char(3)	LAUTH1, swu, coded (000=no, 001=yes)
l2	char(3)	LAUTH2, swu, coded (000=no, 001=yes)
l3	char(3)	LAUTH3, swu, coded (000=no, 001=yes)
l4	char(3)	LAUTH4, swu, coded (000=no, 001=yes)
l5	char(3)	LAUTH5, swu, coded (000=no, 001=yes)
l6	char(3)	LAUTH6, swu, coded (000=no, 001=yes)
l7	char(3)	LAUTH7, swu, coded (000=no, 001=yes)
l8	char(3)	LAUTH8, swu, coded (000=no, 001=yes)
l9	char(3)	LAUTH9, swu, coded (000=no, 001=yes)
l10	char(3)	LAUTH10, swu, coded (000=no, 001=yes)
l11	char(3)	LAUTH11, swu, coded (000=no, 001=yes)
l12	char(3)	LAUTH12, swu, coded (000=no, 001=yes)
l13	char(3)	LAUTH13, swu, coded (000=no, 001=yes)

l14	char(3)	LAUTH14, swu, coded (000=no, 001=yes)
l15	char(3)	LAUTH15, swu, coded (000=no, 001=yes)
l16	char(3)	LAUTH16, swu, coded (000=no, 001=yes)
l17	char(3)	LAUTH17, swu, coded (000=no, 001=yes)
l18	char(3)	LAUTH18, swu, coded (000=no, 001=yes)
l19	char(3)	LAUTH19, swu, coded (000=no, 001=yes)
l20	char(3)	LAUTH20, swu, coded (000=no, 001=yes)
l21	char(3)	LAUTH21, swu, coded (000=no, 001=yes)
l22	char(3)	LAUTH22, swu, coded (000=no, 001=yes)
l23	char(3)	LAUTH23, swu, coded (000=no, 001=yes)
l24	char(3)	LAUTH24, swu, coded (000=no, 001=yes)
l25	char(3)	LAUTH25, swu, coded (000=no, 001=yes)
l26	char(3)	LAUTH26, swu, coded (000=no, 001=yes)
l27	char(3)	LAUTH27, swu, coded (000=no, 001=yes)
l28	char(3)	LAUTH28, swu, coded (000=no, 001=yes)
l29	char(3)	LAUTH29, swu, coded (000=no, 001=yes)
l30	char(3)	LAUTH30, swu, coded (000=no, 001=yes)
l31	char(3)	LAUTH31, swu, coded (000=no, 001=yes)
l32	char(3)	LAUTH32, swu, coded (000=no, 001=yes)
l33	char(3)	LAUTH33, swu, coded (000=no, 001=yes)
l34	char(3)	LAUTH34, swu, coded (000=no, 001=yes)
l35	char(3)	LAUTH35, swu, coded (000=no, 001=yes)
l36	char(3)	LAUTH36, swu, coded (000=no, 001=yes)
l37	char(3)	LAUTH37, swu, coded (000=no, 001=yes)
l38	char(3)	LAUTH38, swu, coded (000=no, 001=yes)
l39	char(3)	LAUTH39, swu, coded (000=no, 001=yes)
l40	char(3)	LAUTH40, swu, coded (000=no, 001=yes)
l41	char(3)	LAUTH41, swu, coded (000=no, 001=yes)
l42	char(3)	LAUTH42, swu, coded (000=no, 001=yes)
l43	char(3)	LAUTH43, swu, coded (000=no, 001=yes)
l44	char(3)	LAUTH44, swu, coded (000=no, 001=yes)
l45	char(3)	LAUTH45, swu, coded (000=no, 001=yes)

l46	char(3)	LAUTH46, swu, coded (000=no, 001=yes)
l47	char(3)	LAUTH47, swu, coded (000=no, 001=yes)
l48	char(3)	LAUTH48, swu, coded (000=no, 001=yes)
l49	char(3)	LAUTH49, swu, coded (000=no, 001=yes)
l50	char(3)	LAUTH50, swu, coded (000=no, 001=yes)
l51	char(3)	LAUTH51, swu, coded (000=no, 001=yes)
l52	char(3)	LAUTH52, swu, coded (000=no, 001=yes)
l53	char(3)	LAUTH53, swu, coded (000=no, 001=yes)
l54	char(3)	LAUTH54, swu, coded (000=no, 001=yes)
l55	char(3)	LAUTH55, swu, coded (000=no, 001=yes)
l56	char(3)	LAUTH56, swu, coded (000=no, 001=yes)
l57	char(3)	LAUTH57, swu, coded (000=no, 001=yes)
l58	char(3)	LAUTH58, swu, coded (000=no, 001=yes)
l59	char(3)	LAUTH59, swu, coded (000=no, 001=yes)
l60	char(3)	LAUTH60, swu, coded (000=no, 001=yes)
l61	char(3)	LAUTH61, swu, coded (000=no, 001=yes)
l62	char(3)	LAUTH62, swu, coded (000=no, 001=yes)
l63	char(3)	LAUTH63, swu, coded (000=no, 001=yes)
l64	char(3)	LAUTH64, swu, coded (000=no, 001=yes)
lcr_opt1	char(3)	lcr_opt1, swu, coded (000=no, 001=yes)
lcr_opt2	char(3)	lcr_opt2, swu, coded (000=no, 001=yes)
lcr_opt3	char(3)	lcr_opt3, swu, coded (000=no, 001=yes)
lcr_opt4	char(3)	lcr_opt4, swu, coded (000=no, 001=yes)
lcr_opt5	char(3)	lcr_opt5, swu, coded (000=no, 001=yes)
lcr_opt6	char(3)	lcr_opt6, swu, coded (000=no, 001=yes)
lcr_opt7	char(3)	lcr_opt7, swu, coded (000=no, 001=yes)
sync_by	char(4)	copied from
sync_time	int	on
info2	char(50)	AMO Description, swu

10.2.1.54 LCR

Table 77: LCR

LCR		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
lrtg	char(5)	LCR routing, swu, key
lrtg_name	char(16)	name, swu
trunk_group	char(3)	Trunk group number, swu
lcrrule	char(4)	Outdial rule, swu
lwert	int	Priority, swu
pos	int	sequ_num, swu, key
carrier_nr	char(2)	Network carrier, swu
carrier_name	char(50)	Network carrier name, swu
zone	char(3)	Zone, swu, coded (see lcr, zone)
lcr_attr_0	char(3)	NONE, swu, coded (000=no, 001=yes)
voice	char(3)	Language, swu, key, coded (000=no, 001=yes)
data	char(3)	Data, swu, key, coded (000=no, 001=yes)
both	char(3)	All, swu, key, coded (000=no, 001=yes)
sched_a	char(3)	Blocked A, swu, coded (000=no, 001=yes)
sched_b	char(3)	Blocked B, swu, coded (000=no, 001=yes)
sched_c	char(3)	Blocked C, swu, coded (000=no, 001=yes)
sched_d	char(3)	Blocked D, swu, coded (000=no, 001=yes)
sched_e	char(3)	Blocked E, swu, coded (000=no, 001=yes)
sched_f	char(3)	Blocked F, swu, coded (000=no, 001=yes)
sched_g	char(3)	Blocked G, swu, coded (000=no, 001=yes)
sched_h	char(3)	Blocked H, swu, coded (000=no, 001=yes)
sched_ae	char(3)	Expensive A, swu, coded (000=no, 001=yes)
sched_be	char(3)	Expensive B, swu, coded (000=no, 001=yes)
sched_ce	char(3)	Expensive C, swu, coded (000=no, 001=yes)

sched_de	char(3)	Expensive D, swu, coded (000=no, 001=yes)
sched_ee	char(3)	Expensive E, swu, coded (000=no, 001=yes)
sched_fe	char(3)	Expensive F, swu, coded (000=no, 001=yes)
sched_ge	char(3)	Expensive G, swu, coded (000=no, 001=yes)
sched_he	char(3)	Expensive H, swu, coded (000=no, 001=yes)
switch_node	char(10)	nodenum, swu
dest_node	char(10)	Dest. node number, swu
lcr_attr_1	char(3)	TRANSGR, swu, coded (000=no, 001=yes)
lcr_attr_2	char(3)	NATNUM, swu, coded (000=no, 001=yes)
lcr_attr_3	char(3)	INTERNUM, swu, coded (000=no, 001=yes)
lcr_attr_4	char(3)	TLNNUM, swu, coded (000=no, 001=yes)
lcr_attr_5	char(3)	NUMPLAN, swu, coded (000=no, 001=yes)
lcr_attr_6	char(3)	WCHREG, swu, coded (000=no, 001=yes)
lcr_attr_7	char(3)	NOTRANS, swu, coded (000=no, 001=yes)
lcr_attr_8	char(3)	NOCHDISP, swu, coded (000=no, 001=yes)
lcr_attr_9	char(3)	INTCHARG, swu, coded (000=no, 001=yes)
lcr_attr_10	char(3)	EODSTCDR, swu, coded (000=no, 001=yes)
lcr_attr_11	char(3)	MODEM, swu, coded (000=no, 001=yes)
lcr_attr_12	char(3)	SUPPCPN, swu, coded (000=no, 001=yes)
lcr_attr_13	char(3)	VOICO, swu, coded (000=no, 001=yes)
lcr_attr_14	char(3)	TGPUB, swu, coded (000=no, 001=yes)
lcr_attr_15	char(3)	TGPRIV, swu, coded (000=no, 001=yes)
lcr_attr_16	char(3)	Reserve 3, swu, coded (000=no, 001=yes)
lcr_attr_17	char(3)	Reserve 4, swu, coded (000=no, 001=yes)
lcr_attr_18	char(3)	Reserve 5, swu, coded (000=no, 001=yes)
lcr_attr_19	char(3)	Reserve 6, swu, coded (000=no, 001=yes)
lcr_attr_20	char(3)	Reserve 7, swu, coded (000=no, 001=yes)
lcr_attr_21	char(3)	Reserve 8, swu, coded (000=no, 001=yes)
lcr_attr_22	char(3)	Reserve 9, swu, coded (000=no, 001=yes)
lcr_attr_23	char(3)	Reserve 10, swu, coded (000=no, 001=yes)
lcr_attr_24	char(3)	lcr_attr_24, swu, coded (000=no, 001=yes)
lcr_attr_25	char(3)	lcr_attr_25, swu, coded (000=no, 001=yes)

lcr_attr_26	char(3)	lcr_attr_26, swu, coded (000=no, 001=yes)
lcr_attr_27	char(3)	lcr_attr_27, swu, coded (000=no, 001=yes)
lcr_attr_28	char(3)	lcr_attr_28, swu, coded (000=no, 001=yes)
lcr_attr_29	char(3)	lcr_attr_29, swu, coded (000=no, 001=yes)
lcr_attr_30	char(3)	lcr_attr_30, swu, coded (000=no, 001=yes)
lcr_attr_31	char(3)	lcr_attr_31, swu, coded (000=no, 001=yes)
lcr_attr_32	char(3)	lcr_attr_32, swu, coded (000=no, 001=yes)
vccyc	char(2)	Compression cycles, swu
bandw	char(2)	Bandwidth, swu
mso_1	char(3)	H0, swu, coded (000=no, 001=yes)
mso_2	char(3)	H11, swu, coded (000=no, 001=yes)
mso_3	char(3)	H12, swu, coded (000=no, 001=yes)
mso_4	char(3)	MULTIRATE, swu, coded (000=no, 001=yes)
mso_5	char(3)	NONE, swu, coded (000=no, 001=yes)
ldsrt_1	char(3)	ldsrt_1, swu, coded (000=no, 001=yes)
ldsrt_2	char(3)	ldsrt_2, swu, coded (000=no, 001=yes)
ldsrt_3	char(3)	ldsrt_3, swu, coded (000=no, 001=yes)
ldsrt_4	char(3)	ldsrt_4, swu, coded (000=no, 001=yes)
ldsrt_5	char(3)	ldsrt_5, swu, coded (000=no, 001=yes)
ldsrt_6	char(3)	ldsrt_6, swu, coded (000=no, 001=yes)
ldsrt_7	char(3)	ldsrt_7, swu, coded (000=no, 001=yes)
ldsrt_8	char(3)	ldsrt_8, swu, coded (000=no, 001=yes)
ldsrt_9	char(3)	ldsrt_9, swu, coded (000=no, 001=yes)
ldsrt_10	char(3)	ldsrt_10, swu, coded (000=no, 001=yes)
ldsrt_11	char(3)	ldsrt_11, swu, coded (000=no, 001=yes)
ldsrt_12	char(3)	ldsrt_12, swu, coded (000=no, 001=yes)
ldsrt_13	char(3)	ldsrt_13, swu, coded (000=no, 001=yes)
ldsrt_14	char(3)	ldsrt_14, swu, coded (000=no, 001=yes)
ldsrt_15	char(3)	ldsrt_15, swu, coded (000=no, 001=yes)
ldsrt_16	char(3)	ldsrt_16, swu, coded (000=no, 001=yes)
bu_lber_1	char(3)	bu_lber_1, swu, coded (000=no, 001=yes)
bu_lber_2	char(3)	bu_lber_2, swu, coded (000=no, 001=yes)

bu_lber_3	char(3)	bu_lber_3, swu, coded (000=no, 001=yes)
bu_lber_4	char(3)	bu_lber_4, swu, coded (000=no, 001=yes)
bu_lber_5	char(3)	bu_lber_5, swu, coded (000=no, 001=yes)
bu_lber_6	char(3)	bu_lber_6, swu, coded (000=no, 001=yes)
bu_lber_7	char(3)	bu_lber_7, swu, coded (000=no, 001=yes)
bu_lber_8	char(3)	bu_lber_8, swu, coded (000=no, 001=yes)
bu_lber_9	char(3)	bu_lber_9, swu, coded (000=no, 001=yes)
bu_lber_10	char(3)	bu_lber_10, swu, coded (000=no, 001=yes)
bu_lber_11	char(3)	bu_lber_11, swu, coded (000=no, 001=yes)
bu_lber_12	char(3)	bu_lber_12, swu, coded (000=no, 001=yes)
bu_lber_13	char(3)	bu_lber_13, swu, coded (000=no, 001=yes)
bu_lber_14	char(3)	bu_lber_14, swu, coded (000=no, 001=yes)
bu_lber_15	char(3)	bu_lber_15, swu, coded (000=no, 001=yes)
bu_lber_16	char(3)	bu_lber_16, swu, coded (000=no, 001=yes)
bu_lber_17	char(3)	bu_lber_17, swu, coded (000=no, 001=yes)
bu_lber_18	char(3)	bu_lber_18, swu, coded (000=no, 001=yes)
bu_lber_19	char(3)	bu_lber_19, swu, coded (000=no, 001=yes)
bu_lber_20	char(3)	bu_lber_20, swu, coded (000=no, 001=yes)
bu_lber_21	char(3)	bu_lber_21, swu, coded (000=no, 001=yes)
bu_lber_22	char(3)	bu_lber_22, swu, coded (000=no, 001=yes)
bu_lber_23	char(3)	bu_lber_23, swu, coded (000=no, 001=yes)
bu_lber_24	char(3)	bu_lber_24, swu, coded (000=no, 001=yes)
bu_lber_25	char(3)	bu_lber_25, swu, coded (000=no, 001=yes)
bu_lber_26	char(3)	bu_lber_26, swu, coded (000=no, 001=yes)
bu_lber_27	char(3)	bu_lber_27, swu, coded (000=no, 001=yes)
bu_lber_28	char(3)	bu_lber_28, swu, coded (000=no, 001=yes)
bu_lber_29	char(3)	bu_lber_29, swu, coded (000=no, 001=yes)
bu_lber_30	char(3)	bu_lber_30, swu, coded (000=no, 001=yes)
bu_lber_31	char(3)	bu_lber_31, swu, coded (000=no, 001=yes)
bu_lber_32	char(3)	bu_lber_32, swu, coded (000=no, 001=yes)
bu_lber_33	char(3)	bu_lber_33, swu, coded (000=no, 001=yes)
bu_lber_34	char(3)	bu_lber_34, swu, coded (000=no, 001=yes)

bu_lber_35	char(3)	bu_lber_35, swu, coded (000=no, 001=yes)
bu_lber_36	char(3)	bu_lber_36, swu, coded (000=no, 001=yes)
bu_lber_37	char(3)	bu_lber_37, swu, coded (000=no, 001=yes)
bu_lber_38	char(3)	bu_lber_38, swu, coded (000=no, 001=yes)
bu_lber_39	char(3)	bu_lber_39, swu, coded (000=no, 001=yes)
bu_lber_40	char(3)	bu_lber_40, swu, coded (000=no, 001=yes)
bu_lber_41	char(3)	bu_lber_41, swu, coded (000=no, 001=yes)
bu_lber_42	char(3)	bu_lber_42, swu, coded (000=no, 001=yes)
bu_lber_43	char(3)	bu_lber_43, swu, coded (000=no, 001=yes)
bu_lber_44	char(3)	bu_lber_44, swu, coded (000=no, 001=yes)
bu_lber_45	char(3)	bu_lber_45, swu, coded (000=no, 001=yes)
bu_lber_46	char(3)	bu_lber_46, swu, coded (000=no, 001=yes)
bu_lber_47	char(3)	bu_lber_47, swu, coded (000=no, 001=yes)
bu_lber_48	char(3)	bu_lber_48, swu, coded (000=no, 001=yes)
bu_lber_49	char(3)	bu_lber_49, swu, coded (000=no, 001=yes)
bu_lber_50	char(3)	bu_lber_50, swu, coded (000=no, 001=yes)
bu_lber_51	char(3)	bu_lber_51, swu, coded (000=no, 001=yes)
bu_lber_52	char(3)	bu_lber_52, swu, coded (000=no, 001=yes)
bu_lber_53	char(3)	bu_lber_53, swu, coded (000=no, 001=yes)
bu_lber_54	char(3)	bu_lber_54, swu, coded (000=no, 001=yes)
bu_lber_55	char(3)	bu_lber_55, swu, coded (000=no, 001=yes)
bu_lber_56	char(3)	bu_lber_56, swu, coded (000=no, 001=yes)
bu_lber_57	char(3)	bu_lber_57, swu, coded (000=no, 001=yes)
bu_lber_58	char(3)	bu_lber_58, swu, coded (000=no, 001=yes)
bu_lber_59	char(3)	bu_lber_59, swu, coded (000=no, 001=yes)
bu_lber_60	char(3)	bu_lber_60, swu, coded (000=no, 001=yes)
bu_lber_61	char(3)	bu_lber_61, swu, coded (000=no, 001=yes)
bu_lber_62	char(3)	bu_lber_62, swu, coded (000=no, 001=yes)
bu_lber_63	char(3)	bu_lber_63, swu, coded (000=no, 001=yes)
bu_lber_64	char(3)	bu_lber_64, swu, coded (000=no, 001=yes)
cbmattr	char(3)	cbmattr, swu, coded
sccid	char(4)	sccid, swu

facility	char(5)	facility, swu
svcvce	char(3)	svcvce, swu, coded
svcnvce	char(3)	svcnvce, swu, coded
lcr_attr_33	char(3)	end of DMC domain / Ende der DMC domain, coded (000=no, 001=yes
gw1	char(9)	gw1 , swu
gw2	char(9)	gw2 , swu
gw3	char(9)	gw3 , swu
gw4	char(9)	gw4 , swu
gw5	char(9)	gw5 , swu
mlpp_flashov	char(3)	swu, coded
mlpp_flash	char(3)	swu, coded
mlpp_immed	char(3)	swu, coded
mlpp_prty	char(3)	swu, coded
mlpp_rtine	char(3)	swu, coded
mlpp_rtinep	char(3)	swu, coded
mlpp_none	char(3)	swu, coded

10.2.1.55 LCRAUTH

Table 78: LCRAUTH

LCRAUTH		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
vbz	char(2)	ITR, swu, key
index	char(1)	index of authorisation code, swu, key
info	char(300)	info
format	char(3)	code format, swu, coded
code	char(22)	authorization code, swu

10.2.1.56 LCRDPLN

Table 79: LCRDPLN

LCRDPLN		
:EXPORT:SELECT		
unique_key	int	
switch_name	char(4)	key
domain	char(8)	
pattern	char(32)	swu, key
info	char(300)	
ips	char(2)	swu
switch_node	char(10)	swu
number	char(32)	swu
length	int	swu
dialplan_num	char(4)	dialplan number, swu
pool_num	char(1)	pool number, swu
service_info	char(40)	service information, swu
dialpln	char(3)	dialplan,swu

10.2.1.57 LCRDPLN_SUB

Table 80: LCRDPLN_SUB

LCRDPLN_SUB		
:EXPORT:SELECT		
unique_key	int	
switch_name	char(4)	key
domain	char(8)	
pattern	char(32)	swu, key
lrtg	char(5)	swu
wabe	char(2)	swu, key
switch_node	char(10)	swu
sync_by	char(4)	
sync_time	int	

fdsidx	char(2)	swu
sdsidx	char(2)	swu
facpinrq	char(3)	swu, coded (000=no, 001=yes)
lber_1	char(3)	swu, coded (000=no, 001=yes)
lber_2	char(3)	swu, coded (000=no, 001=yes)
lber_3	char(3)	swu, coded (000=no, 001=yes)
lber_4	char(3)	swu, coded (000=no, 001=yes)
lber_5	char(3)	swu, coded (000=no, 001=yes)
lber_6	char(3)	swu, coded (000=no, 001=yes)
lber_7	char(3)	swu, coded (000=no, 001=yes)
lber_8	char(3)	swu, coded (000=no, 001=yes)
lber_9	char(3)	swu, coded (000=no, 001=yes)
lber_10	char(3)	swu, coded (000=no, 001=yes)
lber_11	char(3)	swu, coded (000=no, 001=yes)
lber_12	char(3)	swu, coded (000=no, 001=yes)
lber_13	char(3)	swu, coded (000=no, 001=yes)
lber_14	char(3)	swu, coded (000=no, 001=yes)
lber_15	char(3)	swu, coded (000=no, 001=yes)
lber_16	char(3)	swu, coded (000=no, 001=yes)
lber_17	char(3)	swu, coded (000=no, 001=yes)
lber_18	char(3)	swu, coded (000=no, 001=yes)
lber_19	char(3)	swu, coded (000=no, 001=yes)
lber_20	char(3)	swu, coded (000=no, 001=yes)
lber_21	char(3)	swu, coded (000=no, 001=yes)
lber_22	char(3)	swu, coded (000=no, 001=yes)
lber_23	char(3)	swu, coded (000=no, 001=yes)
lber_24	char(3)	swu, coded (000=no, 001=yes)
lber_25	char(3)	swu, coded (000=no, 001=yes)
lber_26	char(3)	swu, coded (000=no, 001=yes)
lber_27	char(3)	swu, coded (000=no, 001=yes)
lber_28	char(3)	swu, coded (000=no, 001=yes)
lber_29	char(3)	swu, coded (000=no, 001=yes)

lber_30	char(3)	swu, coded (000=no, 001=yes)
lber_31	char(3)	swu, coded (000=no, 001=yes)
lber_32	char(3)	swu, coded (000=no, 001=yes)
lber_33	char(3)	swu, coded (000=no, 001=yes)
lber_34	char(3)	swu, coded (000=no, 001=yes)
lber_35	char(3)	swu, coded (000=no, 001=yes)
lber_36	char(3)	swu, coded (000=no, 001=yes)
lber_37	char(3)	swu, coded (000=no, 001=yes)
lber_38	char(3)	swu, coded (000=no, 001=yes)
lber_39	char(3)	swu, coded (000=no, 001=yes)
lber_40	char(3)	swu, coded (000=no, 001=yes)
lber_41	char(3)	swu, coded (000=no, 001=yes)
lber_42	char(3)	swu, coded (000=no, 001=yes)
lber_43	char(3)	swu, coded (000=no, 001=yes)
lber_44	char(3)	swu, coded (000=no, 001=yes)
lber_45	char(3)	swu, coded (000=no, 001=yes)
lber_46	char(3)	swu, coded (000=no, 001=yes)
lber_47	char(3)	swu, coded (000=no, 001=yes)
lber_48	char(3)	swu, coded (000=no, 001=yes)
lber_49	char(3)	swu, coded (000=no, 001=yes)
lber_50	char(3)	swu, coded (000=no, 001=yes)
lber_51	char(3)	swu, coded (000=no, 001=yes)
lber_52	char(3)	swu, coded (000=no, 001=yes)
lber_53	char(3)	swu, coded (000=no, 001=yes)
lber_54	char(3)	swu, coded (000=no, 001=yes)
lber_55	char(3)	swu, coded (000=no, 001=yes)
lber_56	char(3)	swu, coded (000=no, 001=yes)
lber_57	char(3)	swu, coded (000=no, 001=yes)
lber_58	char(3)	swu, coded (000=no, 001=yes)
lber_59	char(3)	swu, coded (000=no, 001=yes)
lber_60	char(3)	swu, coded (000=no, 001=yes)
lber_61	char(3)	swu, coded (000=no, 001=yes)

lber_62	char(3)	swu, coded (000=no, 001=yes)
lber_63	char(3)	swu, coded (000=no, 001=yes)
lber_64	char(3)	swu, coded (000=no, 001=yes)
profidx	char(3)	swu
gateway_num	char(4)	gateway number, swu
dialplan_num	char(4)	dialplan number, swu

10.2.1.58 LCRRULE

Table 81: LCRRULE

LCRRULE		
:EXPORT:SELECT		
unique_key	int	
switch_name	char(4)	key
domain	char(8)	
lcrrule	char(4)	swu, key
info	char(300)	
sync_by	char(4)	
sync_time	int	
switch_node	char(10)	swu
info2	char(50)	swu

10.2.1.59 LCRRULE_SUB

Table 82: LCRRULE_SUB

LCRRULE_SUB		
:EXPORT:SELECT		
unique_key	int	
switch_name	char(4)	key
domain	char(8)	
lcrrule	char(4)	swu, key
rule_element	char(3)	swu, coded (see lcrrule_sub, rule_element)

param	char(22)	swu
position	int	swu, key
switch_node	char(10)	swu

10.2.1.60 LCRSCHED

Table 83: LCRSCHED

LCRSCHED		
:EXPORT:SELECT		
unique_key	int	
switch_name	char(4)	key
domain	char(8)	swu
vbz	char(2)	swu, key
switch_node	char(10)	swu
sync_by	char(4)	
sync_time	int	
sun_00	char(3)	swu
sun_01	char(3)	swu
sun_02	char(3)	swu
sun_03	char(3)	swu
sun_04	char(3)	swu
sun_05	char(3)	swu
sun_06	char(3)	swu
sun_07	char(3)	swu
sun_08	char(3)	swu
sun_09	char(3)	swu
sun_10	char(3)	swu
sun_11	char(3)	swu
sun_12	char(3)	swu
sun_13	char(3)	swu
sun_14	char(3)	swu
sun_15	char(3)	swu

sun_16	char(3)	swu
sun_17	char(3)	swu
sun_18	char(3)	swu
sun_19	char(3)	swu
sun_20	char(3)	swu
sun_21	char(3)	swu
sun_22	char(3)	swu
sun_23	char(3)	swu
mon_00	char(3)	swu
mon_01	char(3)	swu
mon_02	char(3)	swu
mon_03	char(3)	swu
mon_04	char(3)	swu
mon_05	char(3)	swu
mon_06	char(3)	swu
mon_07	char(3)	swu
mon_08	char(3)	swu
mon_09	char(3)	swu
mon_10	char(3)	swu
mon_11	char(3)	swu
mon_12	char(3)	swu
mon_13	char(3)	swu
mon_14	char(3)	swu
mon_15	char(3)	swu
mon_16	char(3)	swu
mon_17	char(3)	swu
mon_18	char(3)	swu
mon_19	char(3)	swu
mon_20	char(3)	swu
mon_21	char(3)	swu
mon_22	char(3)	swu
mon_23	char(3)	swu

tue_00	char(3)	swu
tue_01	char(3)	swu
tue_02	char(3)	swu
tue_03	char(3)	swu
tue_04	char(3)	swu
tue_05	char(3)	swu
tue_06	char(3)	swu
tue_07	char(3)	swu
tue_08	char(3)	swu
tue_09	char(3)	swu
tue_10	char(3)	swu
tue_11	char(3)	swu
tue_12	char(3)	swu
tue_13	char(3)	swu
tue_14	char(3)	swu
tue_15	char(3)	swu
tue_16	char(3)	swu
tue_17	char(3)	swu
tue_18	char(3)	swu
tue_19	char(3)	swu
tue_20	char(3)	swu
tue_21	char(3)	swu
tue_22	char(3)	swu
tue_23	char(3)	swu
wed_00	char(3)	swu
wed_01	char(3)	swu
wed_02	char(3)	swu
wed_03	char(3)	swu
wed_04	char(3)	swu
wed_05	char(3)	swu
wed_06	char(3)	swu
wed_07	char(3)	swu

wed_08	char(3)	swu
wed_09	char(3)	swu
wed_10	char(3)	swu
wed_11	char(3)	swu
wed_12	char(3)	swu
wed_13	char(3)	swu
wed_14	char(3)	swu
wed_15	char(3)	swu
wed_16	char(3)	swu
wed_17	char(3)	swu
wed_18	char(3)	swu
wed_19	char(3)	swu
wed_20	char(3)	swu
wed_21	char(3)	swu
wed_22	char(3)	swu
wed_23	char(3)	swu
thu_00	char(3)	swu
thu_01	char(3)	swu
thu_02	char(3)	swu
thu_03	char(3)	swu
thu_04	char(3)	swu
thu_05	char(3)	swu
thu_06	char(3)	swu
thu_07	char(3)	swu
thu_08	char(3)	swu
thu_09	char(3)	swu
thu_10	char(3)	swu
thu_11	char(3)	swu
thu_12	char(3)	swu
thu_13	char(3)	swu
thu_14	char(3)	swu
thu_15	char(3)	swu

thu_16	char(3)	swu
thu_17	char(3)	swu
thu_18	char(3)	swu
thu_19	char(3)	swu
thu_20	char(3)	swu
thu_21	char(3)	swu
thu_22	char(3)	swu
thu_23	char(3)	swu
fri_00	char(3)	swu
fri_01	char(3)	swu
fri_02	char(3)	swu
fri_03	char(3)	swu
fri_04	char(3)	swu
fri_05	char(3)	swu
fri_06	char(3)	swu
fri_07	char(3)	swu
fri_08	char(3)	swu
fri_09	char(3)	swu
fri_10	char(3)	swu
fri_11	char(3)	swu
fri_12	char(3)	swu
fri_13	char(3)	swu
fri_14	char(3)	swu
fri_15	char(3)	swu
fri_16	char(3)	swu
fri_17	char(3)	swu
fri_18	char(3)	swu
fri_19	char(3)	swu
fri_20	char(3)	swu
fri_21	char(3)	swu
fri_22	char(3)	swu
fri_23	char(3)	swu

sat_00	char(3)	swu
sat_01	char(3)	swu
sat_02	char(3)	swu
sat_03	char(3)	swu
sat_04	char(3)	swu
sat_05	char(3)	swu
sat_06	char(3)	swu
sat_07	char(3)	swu
sat_08	char(3)	swu
sat_09	char(3)	swu
sat_10	char(3)	swu
sat_11	char(3)	swu
sat_12	char(3)	swu
sat_13	char(3)	swu
sat_14	char(3)	swu
sat_15	char(3)	swu
sat_16	char(3)	swu
sat_17	char(3)	swu
sat_18	char(3)	swu
sat_19	char(3)	swu
sat_20	char(3)	swu
sat_21	char(3)	swu
sat_22	char(3)	swu
sat_23	char(3)	swu

10.2.1.61 LCRZONE

Table 84: LCRZONE

LCRZONE		
:EXPORT:SELECT		
unique_key	int	
carrier_nr	char(2)	swu, key

carrier_name	char(50)	swu
info	char(300)	
zone	char(3)	swu, key, coded (see lcr, zone)
switch_name	char(4)	key
switch_node	char(10)	swu
domain	char(8)	

10.2.1.62 LOCDAT

Table 85: LOCDAT

LOCDAT		
:EXPORT:SELECT		
:IMPORT:INSERT: DELETE:UPDATE		
unique_key	int	unique_key
location	char(15)	(required) location, key
name	char(20)	(required) longname
address1	char(30)	(required) Address 1
address2	char(30)	Address 2

10.2.1.63 LRTG

Table 86: LRTG

LRTG		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
dest_switch	char(4)	switch_new, swu
switch_node	char(10)	nodenum, swu
dest_node	char(10)	Dest. node number, swu
lrtg	char(5)	Direction, swu, key
lrtg_name	char(16)	name, swu

type	char(3)	Direction type, swu, coded (see lrtg, type)
rerout	char(3)	Rerouting, swu, coded (000=no, 001=yes)
opti	char(3)	Path optimisation, swu, coded (000=no, 001=yes)
info	char(300)	info
umlver	char(3)	Transfer prevention, swu, coded (000=no, 001=yes)
vlver	char(3)	Queuing prevention, swu, coded (000=no, 001=yes)
mfvanz	char(3)	DTMF display, swu, coded (see lcrrule_sub, rule_element)
mfvtext	char(16)	DTMF text, swu
mfvums	char(3)	DTMF conversion, swu, coded (see lrtg, mfvums)
pin	char(12)	pin_num, swu
mfvpuls	char(3)	DTMF Make/Break Ratio, swu, coded (see lrtg, mfvpuls)
noprcfwd	char(3)	swu coded (000=no; 001=yes) no precedence call forwarding / keine Umlenkung von Precedence Anrufen
clnamedl	char(3)	swu, coded
cstaons	char(3)	swu, coded

10.2.1.64 LRTG_SUB

Table 87: LRTG_SUB

LRTG_SUB		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
lrtg	char(5)	Direction, swu, key
trunk_group	char(3)	Trunk group, swu, key
voice	char(3)	Language, swu, coded (000=no, 001=yes)
data	char(3)	Data, swu, coded (000=no, 001=yes)
both	char(3)	All, swu, coded (000=no, 001=yes)
switch_node	char(10)	nodenum, swu

10.2.1.65 NUMBERING_PLAN

Table 88: NUMBERING_PLAN

NUMBERING_PLAN		virtual nodes / Virtuelle Knoten
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	System, key
domain	char(8)	domain
virtual_node_id	char(10)	virtual_node_id, swu, key
partner	char(3)	Partner, swu, coded
default_entry	char(3)	default_entry, swu, coded
replicated_node	char(3)	Active flag represents that virtual node is replicated manually (automatically in the future by feature "FRN9279 (RQ00038228) EON: Improved administration UC integration for VNR networks") by Manager
isdn_cc	char(4)	isdn_cc, swu
isdn_ac	char(7)	isdn_ac, swu
isdn_lc	char(10)	isdn_lc, swu
isdn_ul	char(3)	isdn_ul, swu, coded
isdn_skip_digits	char(2)	isdn_skip_digits, swu
pnp_l2	char(4)	pnp_l2, swu
pnp_l1	char(6)	pnp_l1, swu
pnp_l0	char(10)	pnp_l0, swu
pnp_ul	char(3)	pnp_ul, swu, coded
pnp_skip_digits	char(2)	pnp_skip_digits, swu
virtual_node_num	char(6)	virtual_node_num, swu
skip_digits	char(1)	skip_digits, swu
cac_prefix	char(3)	cac_prefix, swu
info	char(300)	info
sync_by	char(4)	
sync_time	int	
default_wabe	char(2)	swu
wabe0	char(3)	swu, coded

wabe1	char(3)	swu, coded
wabe2	char(3)	swu, coded
wabe3	char(3)	swu, coded
wabe4	char(3)	swu, coded
wabe5	char(3)	swu, coded
wabe6	char(3)	swu, coded
wabe7	char(3)	swu, coded
wabe8	char(3)	swu, coded
wabe9	char(3)	swu, coded
wabe10	char(3)	swu, coded
wabe11	char(3)	swu, coded
wabe12	char(3)	swu, coded
wabe13	char(3)	swu, coded
wabe14	char(3)	swu, coded
wabe15	char(3)	swu, coded
pd	char(6)	
replicated_node	char(3)	swu, coded

10.2.1.66 ORDERS

Table 89: ORDERS

ORDERS		Batch Processing
:EXPORT:SELECT		
o_id	int	order id
o_jb_id	int	order job id
o_user	char(15)	user
o_count	int	count
o_count_ok	int	count ok
o_count_err	int	count error
o_time	char(14)	time
o_mode	char(1)	mode
o_online	char(1)	online

o_flag	char(1)	flag
o_name	char(32)	name
o_last	int	last

10.2.1.67 ORG

Table 90: ORG

ORG		
:EXPORT:SELECT		
:IMPORT:INSERT: DELETE:UPDATE		
unique_key	int	unique_key
country	char(2)	(required) country, key
company	char(30)	(required) company, key
org1	char(40)	Level 1, key
org2	char(40)	Level 2, key
org3	char(40)	Level 3, key
org4	char(40)	org4, key
org5	char(40)	org5, key
org6	char(40)	org6, key
tenant_group	char(20)	Note: **Manager** only

If one of the "org" fields is filled, the previous org field has to be filled, too (except org1).

10.2.1.68 PEN

Table 91: PEN

PEN		
:EXPORT:SELECT		
unique_key	int	unique_key
pen	char(13)	pen, swu, key
switch_name	char(4)	switch_name, key
cardtyp	char(3)	cardtyp, swu, coded (see Boards 300-349)

status	char(3)	Status, swu, coded (see General Coded Values (0-10))
info	char(300)	info
node_1	char(9)	Node 1
subnode_1	char(2)	Subnode 1
line_1	char(3)	Bay 1
strip_1	char(2)	Strip 1
pin_1	char(3)	Pin 1
node_2	char(9)	Node 2
subnode_2	char(2)	Subnode 2
line_2	char(3)	Bay 2
strip_2	char(2)	Pin 2
pin_2	char(3)	pin_2
ctr_for_busy	int	Number of bus stations
domain	char(8)	domain
reserved	char(3)	Reserved, coded (000=no, 001=yes)
extension	char(12)	extension
pen_num	int	pen_num, swu
ltu	char(2)	ltu
lin1	char(22)	lin1, swu
lin2	char(22)	lin2, swu
res_sb_name	char(10)	Reserved for sched. Batch
reserved_time	int	Reservation expires at
origin	char(1)	origin
pen_num_board	int	pen_num_board
ltg	char(1)	swu
slot	char(3)	swu
circuit	char(4)	swu
board_present	char(3)	board_present, swu, coded
tenant_group	char(20)	Note: **Manager** only

10.2.1.69 PERSDAT

Table 92: PERSDAT

PERSDAT :EXPORT:SELECT :SELECT_UPDATES :IMPORT:INSERT: DELETE:UPDATE		
unique_key	int	unique_key, key
country	char(2)	country
company	char(30)	company
org1	char(40)	Org 1
org2	char(40)	Org 2
org3	char(40)	Org 3
org4	char(40)	Org 4
org5	char(40)	Org 5
org6	char(40)	Org 6
name	char(50)	(required) name
sortname	char(60)	sortname
christianname	char(32)	First name
title	char(15)	title
salutation	char(15)	salutation
location	char(15)	location
building	char(30)	Building
longname	char(15)	longname
extension	char(12)	extension
switch_name	char(4)	switch_name
sequ_num	int	sequ_num
tienum	char(6)	tienum
ext_areacode	char(12)	ext_areacode
ext_netcode	char(16)	ext_netcode
ext_ext	char(23)	id

faxnumber	char(22)	faxnumber
import_dat	char(1)	import_dat
owner_group	char(30)	owner_group
modify_date	char(14)	Modification date
access_control	char(1)	access_control
text1	char(50)	text1
text2	char(50)	text2
text3	char(50)	text3
direct_trunk_acc	char(1)	Direct CO Trunk Line
equipment	char(160)	Device Configuration
l1	char(4)	Service Type 1
l2	char(4)	Service Type 2
l3	char(4)	Service Type 3
l4	char(4)	Service Type 4
l5	char(4)	Service Type 5
l6	char(4)	Service Type 6
l7	char(4)	Service Type 7
l8	char(4)	Service Type 8
l9	char(4)	Service Type 9
l10	char(4)	Service Type 10
l11	char(4)	Service Type 11
l12	char(4)	Service Type 12
l13	char(4)	Service Type 13
l14	char(4)	Service Type 14
l15	char(4)	Service Type 15
l16	char(4)	Service Type 16
l17	char(4)	Service Type 17
l18	char(4)	Service Type 18
l19	char(4)	Service Type 19
l20	char(4)	Service Type 20
l_prefix	char(7)	Service type prefix
text4	char(50)	Keywords4

ext_countrycode	char(4)	Ext. Country Code
chargeid	char(32)	chargeid
basic_charge_class	char(12)	Basic Charge Class
bankcode	char(15)	Bank Sort Code
accountnum	char(15)	Account Number
xpr_exists	char(3)	mailbox, coded (see Values for OpenScape Xpressions (Xpressions 450))
xpr_key	char(25)	Mailbox ID
xpr_pin	char(12)	pin_num
xpr_server	char(25)	XPR Server
xpr_vmsadmin	char(3)	Administrator, coded (000=no, 001=yes)
xpr_broadcast	char(3)	Create broadcast messages, coded (000=no, 001=yes)
xpr_dct_send	char(3)	Create dictation, coded (000=no, 001=yes)
xpr_dct_receive	char(3)	Receive dictation, coded (000=no, 001=yes)
xpr_password	char(3)	Passwort required, coded (000=no, 001=yes)
xpr_name	char(3)	Name announcement required, coded (000=no, 001=yes)
xpr_flag1	char(3)	No Company Announcement, coded (000=no, 001=yes)
xpr_flag2	char(3)	Prompt 'Infobox reached', coded (000=no, 001=yes)
xpr_connect	char(3)	Connection Rights, coded (see Values for OpenScape Xpressions (Xpressions 450))
xpr_quota	char(12)	Quota
xpr_reserve1	char(12)	Delegate
xpr_reserve2	char(100)	Customer-specific 4
persnum	char(32)	persnum
domain	char(8)	domain
idf	char(10)	IDF
idf_cable_pair	char(10)	Cable pair
info_5	char(50)	Info 5
info_6	char(50)	Info 6
info_7	char(50)	Info 7
info_8	char(50)	Info 8
info_9	char(50)	Info 9

info_10	char(50)	Info 10
info_11	char(50)	Info 11
info_12	char(50)	Info 12
info_13	char(50)	Info 13
info_14	char(50)	Info 14
info_15	char(50)	Info 15
info_16	char(50)	Info 16
info_17	char(50)	info13
info_18	char(50)	info14
info_19	char(50)	info15
v_switch	char(4)	virtual switch name
usid	char(40)	usid
umuid	char(50)	umuid
um_user	char(3)	um_user, coded (000=no, 001=yes)
um_marked	char(3)	um_marked, coded (000=no, 001=yes)
um_operation_id	char(50)	um_operation_id
ape_group_no	char(2)	ape group number, swu
xpr_delpwd	char(3)	xpr delete password, coded (000/001)
xpr_delpwd_tmp	char(3)	xpr delete password tmp, coded(000/001)
tenant_group	char(20)	Note: **Manager** only
e_mail_addr	char(320)	
subscr_key	int	
newimpl	char(3)	coded

10.2.1.70 PERSDAT_VNR

Table 93: PERSDAT_VNR

PERSDAT_VNR		
:EXPORT:SELECT		
unique_key	int	unique_key, key
country	char(2)	country
company	char(30)	company

org1	char(40)	Org 1
org2	char(40)	Org 2
org3	char(40)	Org 3
org4	char(40)	Org 4
org5	char(40)	Org 5
org6	char(40)	Org 6
name	char(50)	(required) name
sortname	char(60)	sortname
christianname	char(32)	First name
title	char(15)	title
salutation	char(15)	salutation
location	char(15)	location
building	char(30)	Building
longname	char(15)	longname
extension	char(12)	extension
switch_name	char(4)	switch_name
sequ_num	int	sequ_num
tienum	char(6)	tienum
ext_areacode	char(12)	ext_areacode
ext_netcode	char(16)	ext_netcode
ext_ext	char(23)	id
faxnumber	char(22)	faxnumber
import_dat	char(1)	import_dat
owner_group	char(30)	owner_group
modify_date	char(14)	Modification date
access_control	char(1)	access_control
text1	char(50)	text1
text2	char(50)	text2
text3	char(50)	text3
direct_trunk_acc	char(1)	Direct CO Trunk Line
equipment	char(160)	Device Configuration
l1	char(4)	Service Type 1

l2	char(4)	Service Type 2
l3	char(4)	Service Type 3
l4	char(4)	Service Type 4
l5	char(4)	Service Type 5
l6	char(4)	Service Type 6
l7	char(4)	Service Type 7
l8	char(4)	Service Type 8
l9	char(4)	Service Type 9
l10	char(4)	Service Type 10
l11	char(4)	Service Type 11
l12	char(4)	Service Type 12
l13	char(4)	Service Type 13
l14	char(4)	Service Type 14
l15	char(4)	Service Type 15
l16	char(4)	Service Type 16
l17	char(4)	Service Type 17
l18	char(4)	Service Type 18
l19	char(4)	Service Type 19
l20	char(4)	Service Type 20
l_prefix	char(7)	
text4	char(50)	
ext_countrycode	char(4)	
chargeid	char(32)	
basic_charge_class	char(12)	
bankcode	char(15)	
accountnum	char(15)	
xpr_exists	char(3)	
xpr_key	char(25)	
xpr_pin	char(12)	
xpr_server	char(25)	
xpr_vmsadmin	char(3)	
xpr_broadcast	char(3)	

xpr_dct_send	char(3)
xpr_dct_receive	char(3)
xpr_password	char(3)
xpr_name	char(3)
xpr_flag1	char(3)
xpr_flag2	char(3)
xpr_connect	char(3)
xpr_quota	char(12)
xpr_reserve1	char(12)
xpr_reserve2	char(100)
persnum	char(32)
domain	char(8)
idf	char(10)
idf_cable_pair	char(10)
info_5	char(50)
info_6	char(50)
info_7	char(50)
info_8	char(50)
info_9	char(50)
info_10	char(50)
info_11	char(50)
info_12	char(50)
info_13	char(50)
info_14	char(50)
info_15	char(50)
info_16	char(50)
info_17	char(50)
info_18	char(50)
info_19	char(50)
v_switch	char(4)
usid	char(40)
umuid	char(50)

um_user	char(3)	
um_marked	char(3)	
um_operation_id	char(50)	
ape_group_no	char(2)	
xpr_delpwd	char(3)	
xpr_delpwd_tmp	char(3)	
tenant_group	char(20)	Note: **Manager** only
e_mail_addr	char(320)	
extension_short	char(6)	
vnac	char(6)	
isdn_cc	char(4)	
isdn_ac	char(6)	
isdn_lc	char(10)	

10.2.1.71 PORTPERS/PERSPORT

Table 94: PORTPERS/PERSPORT

ODF name:	**PORTPERS (**Join between PERSDAT/PORT/DISPLAYNAME, Persdat dominant**)**	
Operations:	PERSPORT **(**Join between PERSDAT/PORT/DISPLAYNAME, Port dominant only available for DLL/Export Table/api2hipath**)**	
	Export: **Select** Import: **-**	
Field	Data type	Description
unique_key	int	unique_key
extension	char(12)	extension, swu, key
switch_name	char(4)	switch_name, key
pen	char(13)	pen, swu
devconame	char(20)	Device combination
opstate	char(3)	operation state, swu, coded (000=no, 001=yes, 008=card, 056=logged off)
chargeid	char(6)	chargeid
wabegrpnum	char(2)	wabegrpnum, swu
vbzgrpnum	char(2)	vbzgrpnum, swu
tcos	char(4)	Cos 1, swu

ncos	char(4)	Cos 2, swu
berumgrpnum	char(2)	berumgrpnum, swu
stbnum	char(3)	num_key_lay, swu
bkrnl	char(2)	Speed dial list 1, swu
speed_dial_two	char(2)	Speed dial list 2, swu
speed_dial	char(3)	speed_dial, swu, coded
numdisp	char(3)	displ, swu, coded
devtyp	char(3)	device type, swu, coded
devtext	char(20)	devtext, swu
att_tone_source	char(3)	VC warn tone source, swu, coded
att_tone_target	char(3)	VC warn tone destination, swu, coded
technical_ext	char(6)	Technical extension, swu
switch_new	char(4)	switch_new, swu
temporary	char(3)	Temporary extension, coded
call_back_free	char(3)	Callback-no answer rejection, swu, coded
call_back_busy	char(3)	Callback-busy rejection, swu, coded
displayname	char(31)	Display name, swu, from display_name table
dl_idx_man	char(3)	Manual downloading, swu
dl_idx_auto	char(3)	Automatic downloading, swu
domain	char(8)	domain
least_cost_rout1	char(2)	LCRCOS 1, swu
least_cost_rout2	char(2)	LCRCOS 2, swu
least_cost_data1	char(2)	LCRCOSDAT 1, swu
least_cost_data2	char(2)	LCRCOSDAT 2, swu
navar	char(3)	navar, swu, coded
assoc	char(3)	Associated subscriber, swu, coded
analog_modem	char(3)	analogue_modem, swu, coded
clc_num	char(3)	CLC number, swu, coded
bus_type	char(3)	bus_type, swu, coded
bus_extension	char(6)	bus_extension, swu
tsi_voice	char(1)	TSI voice, swu
tsi_data	char(1)	TSI data, swu

tsi_fax	char(1)	TSI fax, swu
protocol_voice	char(3)	Protocol voice, swu, coded
protocol_data	char(3)	Protocol data, swu, coded
protocol_fax	char(3)	Protocol fax, swu, coded
msn	char(3)	msn, swu, coded
tsi_ttx	char(1)	TSI ttx, swu
tsi_btx	char(1)	TSI vtx, swu
tsi_msn	char(1)	TSI msn, swu
connect_type	char(3)	connect_type, swu, coded
opt_voice	char(2)	Voice option, swu
opt_data	char(2)	Data option, swu
opt_fax	char(2)	Fax option, swu
secr_device	char(3)	Secretary telephone, swu, coded
info	char(50)	info
fcfw	char(22)	fcfw, swu
compl_mod_allowed	char(3)	compl_mod_allowed, swu, coded
registration	char(3)	Registration, swu, coded
opt_btx	char(2)	VTX option, swu
opt_ttx	char(2)	TTX option, swu
prot_btx	char(3)	VTX protocol, swu, coded
prot_ttx	char(3)	TTX protocol, swu, coded
p_akl	char(3)	KN, swu, coded
p_aufaks	char(3)	RKNOVR, swu, coded
p_aufs	char(3)	KNOVR, swu, coded
p_dirans	char(3)	VC, swu, coded
p_hot	char(3)	HOT, swu, coded
p_natr	char(3)	EOVR, swu, coded
p_roe	char(3)	OFHKRC, swu, coded
p_agent	char(3)	AGENT, swu, coded
p_autolog	char(3)	AUTOLOG, swu, coded
p_caf	char(3)	CAF, swu, coded
p_castmon	char(3)	CASTMON, swu, coded

p_cbstmon	char(3)	CBSTMON, swu, coded
p_dndsr	char(3)	DNDSR, swu, coded
p_super	char(3)	SUPER, swu, coded
p_ani1	char(3)	ANI1, swu, coded
p_ani2	char(3)	ANI2, swu, coded
p_ani3	char(3)	ANI3, swu, coded
p_ani4	char(3)	ANI4, swu, coded
p_ani5	char(3)	ANI5, swu, coded
p_ani6	char(3)	ANI6, swu, coded
p_ani7	char(3)	ANI7, swu, coded
p_ani8	char(3)	ANI8, swu, coded
p_ani9	char(3)	ANI9, swu, coded
p_ani10	char(3)	ANI10, swu, coded
pen_num	int	pen_num, swu
net_config	char(3)	Network-Wide Configuration, coded
log_pty_switch	char(4)	Logical Subscriber System, swu
log_pty_ext	char(6)	Logical Subscriber Number, swu
log_pty_pen	char(13)	Logical Subscriber PEN, swu
log_pty_pen_num	int	log_pty_pen_num, swu
rem_pty_switch	char(4)	Remote Subscriber System, swu
rem_pty_ext	char(6)	Remote Subscriber Number, swu
rem_pty_pen	char(13)	Remote Subscriber PEN, swu
rem_pty_pen_num	int	rem_pty_pen_num, swu
extend_pty_ext	char(6)	Extended Connection Number, swu
ext_pty_ext_l	char(22)	Extension number of extended call, swu
ext_pty_ptnr_ext	char(6)	Destination Data Number
ext_pty_ptnr_ext_l	char(22)	Destination Data Extension Number, swu
ext_pty_ptnr_swi	char(4)	Destination Data System
conf_not_consist	char(3)	conf_not_consist, swu, coded
opt_ntvlv	char(2)	NTVLV Option, swu
prot_ntvlv	char(3)	NTVLV Protocol, swu, coded
permakt	char(3)	permakt, swu, coded

cordless	char(3)	cordless, swu, coded
lwpar	char(2)	lwpar, swu
uart	char(3)	uart, swu, coded
num_callback	char(2)	Max. Number of Callbacks, swu
protocol_spp	char(3)	Protocol S0PP/S2PP, swu, coded
opt_spp	char(2)	Option S0PP/S2PP, swu
alarm	char(4)	Alarm Number, swu
circidx	char(3)	CIRCIDX, swu
cofidx	char(2)	COFIDX, swu
call_free_mail	char(3)	Callback on No Answer, swu, coded
language	char(3)	Preferred Language, swu, coded
p_sprstat	char(3)	Language Static, swu, coded
p_sprdyn	char(3)	Language Dynamic, swu, coded
vpi	char(1)	VPI, swu
vci	char(5)	VCI, swu
status	char(26)	Status, swu
call_log	char(3)	Call Logging, swu, coded (A06:none, C18:all, 011:attempts)
api	char(3)	Work Station Protocol, swu, coded
p_spawh	char(3)	LOADF, swu, coded
p_spsc	char(3)	LOSC, swu, coded
p_spenb	char(3)	LOENB, swu, coded
auto_privacy	char(3)	Automatic Privacy, swu, coded
single_button_mode	char(3)	Single Key Mode, swu, coded
out_line_sel	char(3)	Outgoing Call Preference, swu, coded
inc_line_sel	char(3)	Incoming Call Preference, swu, coded
ring_type	char(3)	Subscriber busy warning tone, swu, coded
key_system	char(3)	Key System, swu, coded
busindex	int	busindex, swu
p_teamchef	char(3)	Executive, swu, coded
p_teamsecr	char(3)	Secretary, swu, coded
p_drziel	char(3)	DSS Dest. networkwide, swu, coded

teamnum	char(3)	Exec./Secr. Team, swu
teamfunct	char(3)	Team Function, swu, coded
store_calls	char(3)	Store calls, swu, coded
dtb_access	char(3)	DTB Access, swu, coded
dual_mode_skey	char(3)	Dual Mode Softkey, swu, coded
sms_feature	char(3)	SMS, swu, coded
reg_counter	char(1)	Registration counter, swu
ic_data_type	char(3)	Data Type, swu, coded
ltu	char(2)	ltu
trace_mal	char(3)	malicious call trace allowed, swu, coded
asynct	char(4)	asynchronious time, swu
offtype	char(3)	switching off type, swu, coded
hmusic	char(3)	music on hold recording, swu
extbus	char(3)	extented bus, swu, coded
pattern	char(3)	idlecode pattern for the b-channels, swu
pulslev	char(3)	level of charge metering pulses, swu, coded
pulstype	char(3)	type of charge metering pulses, swu, coded
ackst	char(3)	acknowledge state between dps and announcement device, swu, coded
exact_dev_typ	char(3)	exact device type, swu, coded
exact_dev_text	char(20)	remarks to exact device type, swu
ccmscd	char(6)	multiple code-calling code, swu
ccno	char(6)	code-calling number, swu
spec_outpulse_dev	char(3)	spec_outpulse_dev, swu, coded
special	char(3)	special station, swu, coded
sufdial	char(3)	station can suffix dial, swu, coded
witness	char(3)	witness, swu, coded
phone_mail_idx	char(2)	index for PhoneMail / Xpression access number, swu
dnidsp	char(3)	display DNIS text, swu, coded
circuit	char(3)	circuit number
flash_enable	char(3)	enable hook flash, swu, coded
ops_conn_type	char(3)	Off Premise connection type, swu, coded

src_grp_idx	char(2)	source group index, swu
classmrk_ec	char(3)	classmrk_ec, swu, coded
classmrk_g711	char(3)	classmrk_g711, swu, coded
classmrk_g729	char(3)	classmrk_g729, swu, coded
classmrk_vad	char(3)	classmrk_vad, swu, coded
classmrk_init	char(3)	classmrk_init, swu, coded
hotline_idx	char(3)	Hotline Index, swu
node_id	char(10)	node_id, swu
cdracc	char(12)	CDR Account Code, swu
fwd_spk_call	char(3)	fwd_spk_call, swu, coded
p_spk1way	char(3)	p_spk1way, swu, coded
com_group	char(4)	com_group, swu
com_spk_call_2wy	char(3)	com_spk_call_2wy, swu, coded
spk_call_1wy_brd	char(3)	spk_call_1wy_brd, swu, coded
p_elapsed_time	char(3)	Elapsed Time, swu, coded
p_block_auto_rel	char(3)	p_block_auto_rel, swu, coded
protocol_class	char(3)	protocol_class, swu, coded
lin1	char(22)	lin1, swu
lin2	char(22)	lin2, swu
acf_appl	char(3)	acf_appl, swu
dtmf_conf	char(3)	dtmf_conf, swu, coded
dhpar_asend	char(3)	dhpar_asend, swu, coded
dhpar_bsend	char(3)	dhpar_bsend, swu, coded
dhpar_dtmfst	char(3)	dhpar_dtmfst, swu, coded
dhpar_flash	char(3)	dhpar_flash, swu, coded
dhpar_vmsdp	char(3)	dhpar_vmsdp, swu, coded
dhpar_ledsign	char(3)	dhpar_ledsign, swu, coded
dhpar_mwled	char(3)	dhpar_mwled, swu, coded
dhpar_predia	char(3)	dhpar_predia, swu, coded
dhpar_ring1	char(3)	dhpar_ring1, swu, coded
bas_svc_audio31k	char(3)	bas_svc_audio31k, swu, coded
bas_svc_voice	char(3)	bas_svc_voice, swu, coded

bas_svc_udi64	char(3)	bas_svc_udi64, swu, coded
bas_svc_udi64ra	char(3)	bas_svc_udi64ra, swu, coded
sztype_atnd	char(3)	sztype_atnd, swu, coded
sztype_dszco	char(3)	sztype_dszco, swu, coded
sztype_dszint	char(3)	sztype_dszint, swu, coded
sztype_expr	char(3)	sztype_expr, swu, coded
sztype_fwd	char(3)	sztype_fwd, swu, coded
sztype_fwdnans	char(3)	sztype_fwdnans, swu, coded
szcd_atnd	char(4)	szcd_atnd, swu
szcd_dszco	char(4)	szcd_dszco, swu
szcd_dszint	char(4)	szcd_dszint, swu
szcd_expr	char(4)	szcd_expr, swu
szcd_fwd	char(4)	szcd_fwd, swu
szcd_fwdnans	char(4)	szcd_fwdnans, swu
pubnum	char(15)	pubnum, swu
ton	char(3)	ton, swu, coded
npi	char(3)	npi, swu, coded
dtmf_control	char(3)	dtmf_control, swu, coded
delay_cfw_busy	char(3)	delay_cfw_busy, swu, coded
p_cfnr	char(3)	p_cfnr, swu, coded
p_dsfwdext	char(3)	p_dsfwdext, swu, coded
p_mlpp_routine	char(3)	p_mlpp_routine, swu, coded
p_mlpp_prty	char(3)	p_mlpp_prty, swu, coded
p_mlpp_immed	char(3)	p_mlpp_immed, swu, coded
p_mlpp_flash	char(3)	p_mlpp_flash, swu, coded
p_mlpp_flashov	char(3)	p_mlpp_flashov, swu, coded
p_mlpp_prot	char(3)	p_mlpp_prot, swu, coded
p_mlpp_mlpps	char(3)	p_mlpp_mlpps, swu, coded
pen_standard	char(9)	pen_standard
circuit_standard	char(3)	circuit_standard
ippassw	char(20)	ippassw
p_alogontw	char(3)	p_alogontw

classmrk_g729opt	char(3)	classmrk_g729opt
vcepage	char(3)	vcepage
p_reserve_1	char(3)	p_reserve_1
p_reserve_2	char(3)	p_reserve_2
p_reserve_3	char(3)	p_reserve_3
mixed_net_move	char(3)	mixed_net_move
location_code	char(10)	location_code
extension_limit	char(6)	extension_limit
p_dsswoovr	char(3)	p_dsswoovr
um_operation_id	char(50)	um_operation_id
p_dssline	char(3)	p_dssline
p_rngxferl	char(3)	p_rngxferl
p_srnline	char(3)	p_srnline
p_monslntl	char(3)	p_monslntl
display_mode	char(3)	display_mode
lin_temp	char(22)	lin_temp
emerg_no_temp	char(22)	emerg_no_temp
mob_user_via_app	char(3)	mob_user_via_app
p_mhfatbv	char(3)	p_mhfatbv
p_mhfahome	char(3)	p_mhfahome
stno_oos	char(22)	stno_oos
stno_ap_emrgcy	char(22)	stno_ap_emrgcy
mov_hfa_ip_phone	char(3)	mov_hfa_ip_phone
gw_ip_addr	char(15)	gw_ip_addr
dmcallwd	char(3)	dmcallwd
p_mlpp_rtprot	char(3)	p_mlpp_rtprot
mlpp_svcdom	char(19)	mlpp_svcdom
cluster_id	char(3)	cluster_id
p_calltype	char(3)	p_calltype
amoalert	char(3)	amoalert
codec_type	char(3)	codec_type
classmark_hg3530	char(3)	classmark_hg3530

classmark_hg3550	char(3)	classmark_hg3550
ape_group_no_port	char(2)	ape_group_no (from table port)
cmi_handset_type	char(3)	cmi_handset_type
ip_adress	char(3)	ip_adress
country	char(2)	country (from table persdat)
company	char(30)	company (from table persdat)
org1	char(40)	Org 1
org2	char(40)	Org 2
org3	char(40)	Org 3
org4	char(40)	Org 4
org5	char(40)	Org 5
org6	char(40)	Org 6
name	char(50)	name
sortname	char(60)	sortname
christianname	char(30)	christianname
title	char(15)	title
salutation	char(15)	salutation
location	char(15)	location
building	char(30)	Building
longname	char(10)	longname
sequ_num	int	sequ_num
tienum	char(6)	tienum
ext_areacode	char(6)	ext_areacode
ext_netcode	char(16)	ext_netcode
ext_ext	char(6)	id
faxnumber	char(22)	faxnumber
import_dat	char(1)	import_dat
owner_group	char(30)	owner_group
modify_date	char(14)	Modification date
access_control	char(1)	access_control
text1	char(50)	text1
text2	char(50)	text2

text3	char(50)	text3
text4	char(50)	text4
direct_trunk_acc	char(1)	Direct CO Trunk Line
equipment	char(160)	Device Configuration
ext_countrycode	char(4)	externer countrycode
bankcode	char(15)	bankcode
accountnum	char(15)	accountnum
persnum	char(32)	persnum
idf	char(10)	idf
idf_cable_pair	char(10)	idf_cable_pair
v_switch	char(4)	v_switch
usid	char(40)	usid
umuid	char(50)	umuid
um_user	char(3)	um_user
um_marked	char(3)	um_marked
um_operation_id	char(50)	um_operation_id
ape_group_no	char(2)	ape_group_no (from table persdat)
department	char(25)	department (from table display_name)
position	int	position (from table display_name)
charcon	char(3)	charcon (from table display_name)
extension_short	char(6)	
vnac	char(6)	
isdn_cc	char(4)	
isdn_ac	char(6)	
isdn_lc	char(10)	

10.2.1.72 PICKUPGRP

Table 95: PICKUPGRP

PICKUPGRP
:EXPORT:SELECT
:IMPORT:DELETE

unique_key	int	unique_key
pickupgrpnum	char(4)	Pickup, swu, key
switch_name	char(4)	switch_name, key
attcall	char(3)	alerting_call, swu, coded (see Feature 250-269)
displ	char(3)	displ, swu, coded (000=no, 001=yes)
secondcall	char(3)	secondcall, swu, coded (000=no, 001=yes)
info	char(300)	info
domain	char(8)	domain
signal	char(3)	Signal, swu, coded (see lrtg, type)
consistent	char(3)	consistent, swu, coded
teamnum	char(3)	Exec./Secr. Team, swu
fictitious	char(3)	Fictitious PU, swu, coded (000=no, 001=yes)
parallel_sub_ons	char(3)	ONS group - coded yes/no,
num_gsm_phone	char(22)	number of gsm phone
search_mob_first	char(3)	search mobile phone first - coded yes/no
ons_slave_num_sup	char(3)	ONS slave number suppress - coded yes/no
tenant_group	char(20)	name of the tenant group the object belongs to Note: **Manager** only
:IMPORT:INSERT:		
unique_key	int	unique_key
pickupgrpnum	char(4)	Pickup, swu, key
switch_name	char(4)	switch_name, key
attcall	char(3)	alerting_call, swu, coded (see Feature 250-269)
displ	char(3)	displ, swu, coded (000=no, 001=yes)
secondcall	char(3)	secondcall, swu, coded (000=no, 001=yes)
info	char(300)	info
domain	char(8)	domain
signal	char(3)	Signal, swu, coded (see lrtg, type)
consistent	char(3)	consistent, swu, coded
teamnum	char(3)	Exec./Secr. Team, swu
fictitious	char(3)	Fictitious PU, swu, coded (000=no, 001=yes)
parallel_sub_ons	char(3)	ONS group - coded yes/no,

num_gsm_phone	char(22)	number of gsm phone
search_mob_first	char(3)	search mobile phone first - coded yes/no
ons_slave_num_sup	char(3)	ONS slave number suppress - coded yes/no
tenant_group	char(20)	name of the tenant group the object belongs to Note: **Manager** only
extension1	char(12)	extension of the first station in a newly created PICKUPGRP
extension2	char(12)	extension of the second station in a newly created PICKUPGRP

10.2.1.73 PICKUP_LINK

Table 96: PICKUP_LINK

PICKUP_LINK		
:EXPORT:SELECT		
unique_key	int	unique_key
pickupgrpnum	char(4)	Pickup, swu, key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
rem_pickupgrpnum	char(4)	External Call Pickup, swu, key
rem_switch_name	char(4)	External System, swu
signal_an	char(3)	Incoming Signal, swu, coded (000=no, 001=yes)
signal_ab	char(3)	Outgoing Signal, swu, coded (000=no, 001=yes)
signal_park	char(3)	Park Signalling, swu, coded (000=no, 001=yes)
consistent	char(3)	consistent, swu, coded (000=no, 001=yes)
remkz	char(22)	Remote Code, swu, key

10.2.1.74 PICKUP_SUB

Table 97: PICKUP_SUB

PICKUP_SUB		
:EXPORT:SELECT		
unique_key	int	unique_key

pickupgrpnum	char(4)	Pickup, swu, key
switch_name	char(4)	switch_name, key
extension	char(12)	extension, swu, key
domain	char(8)	domain
remote_subscriber	char(3)	Remote Subscriber, swu, coded (000=no, 001=yes)
remote_switch	char(4)	Members System, swu
teamnum	char(3)	Exec./Secr. Team, swu
pararng	char(3)	swu, coded (000=no, 001=yes)
grpcal	char(3)	swu, coded (000=no, 001=yes)
mastergrp	char(3)	swu, coded (000=no, 001=yes)
grpcalb	char(3)	swu, coded (000=no, 001=yes)
template_id	int	unique key of the record in templates table

10.2.1.75 PIN

Table 98: PIN

PIN		
:EXPORT:SELECT: SELECT_UPDATES		
:IMPORT:INSERT: DELETE:UPDATE		
unique_key	int	unique_key
domain	char(8)	domain
extension	char(12)	(required) extension, swu, key
switch_name	char(4)	(required) switch_name, key
pin_num	char(12)	(required) pin_num, swu, key
pin_type	char(3)	pin_type, swu, coded (see PIN Types (41-43), pin,pin_type)
position	int	position
pinindiv	char(3)	Individual PIN, swu, coded (000=no, 001=yes)(default value: 000)
pin_class	char(2)	(required) PIN Class, swu, key
v_switch	char(4)	virtual switch name
um_operation_id	char(50)	um_operation_id

umuid	char(50)	UM User ID
template_id	int	unique key of the record in templates table
subscr_key	int	unique key of the record in persdat table
cordless_bulk_add	char(3)	coded
manuel_bulk_add	char(3)	coded
pen	char(13)	swu

Note: PINs within one switch have to be unique (exception: PINs of the type "berum"). The length of the PIN must match the PIN-length defined for the whole switch. If PIN data are changed, the corresponding PIN must be deleted and set new afterwards.

Overview of PIN classes and PIN types:

class	type	type description	meaning
1	041	Manual PIN	Manual identification
2	G01	Business PIN long	Identification of business calls (remains active until manual deletion or timeout).
3	G02	Business PIN short	Identification of business calls
4	G03	Private PIN	Identification of private calls
5	G04	Project Code PIN	Identification of a project for a call (project code = PCODE).
6	042	ID Card PIN	Identification by means of a magnetic/chip card which is inserted into the card reader of the terminal.
7	043	COS switchover	Identification of users which are authorized to activate COS changeover from the terminal.
8	G05	PIN for Service ID	Pin for service id
9	G07	CDR Account Code	CDR Account Code
10	G06	Forced authorization Code	PIN for dialing pattern check
11	C66	Cordless E PIN	PIN for Cordless Devices

Individual PIN (pinindiv = 001): Each PIN is only assigned to the given PIN type, also if other PIN types of the same length exist.

Non-individual PIN (Feld pinindiv = 000): Each PIN is assigned to the given PIN type and to all other PIN types of the same length (Default is the non-individual PIN, pinindiv = 000).

10.2.1.76 PINCHECK

Table 99: PINCHECK

PINCHECK		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
modulo	char(1)	modulo check algorithm, swu, key
modulus	char(2)	modulus, swu
chknlen	char(1)	length of check number, swu
typechkn	char(3)	type of check number, swu, coded (see pin_check, typechkn)
typesum	char(3)	type of sum of the products, swu, coded (see pin_check, typesum)
valence_1	char(1)	valence_1, swu
valence_2	char(1)	valence_2, swu
valence_3	char(1)	valence_3, swu
valence_4	char(1)	valence_4, swu
valence_5	char(1)	valence_5, swu
valence_6	char(1)	valence_6, swu
valence_7	char(1)	valence_7, swu
valence_8	char(1)	valence_8, swu
valence_9	char(1)	valence_9, swu
valence_10	char(1)	valence_10, swu
valence_11	char(1)	valence_11, swu
valence_12	char(1)	valence_12, swu
info	char(300)	info

10.2.1.77 PINCLASS

Table 100: PINCLASS

PINCLASS

:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name, key
pin_class	char(2)	PIN Class, swu, key
pin_type	char(3)	PIN Type, swu, coded (see PIN Types (41-43), pin,pin_type)
pin_len	char(2)	PIN Length, swu
trkcd	char(6)	trunk access code for automatic dialing, swu
trkcdtyp	char(3)	type of automatic trunk access, swu, coded (see pinclass, trkcdtyp)
disprstn	char(3)	displaying the remote station number to call partner, swu, coded (000=no, 001=yes)
dispid	char(3)	displaying the id, swu, coded (see pinclass, dispid)
disppin	char(3)	displaying the pin, swu, coded (000=no, 001=yes)
coshome	char(3)	use actual cos of home station, swu, coded (000=no, 001=yes)
deackey	char(3)	deactivate pin after cancel or delete key, swu, coded (000=no, 001=yes)
trace_pin	char(3)	trace the pin user, swu, coded (000=no, 001=yes)
cdrtext	char(10)	additional text for call detail recording, swu
cdrtexts	char(3)	send additional text to call detail recording system, swu, coded (000=no, 001=yes)
cotype	char(3)	copin type for network-wide transfer of copin using cornet, swu, coded (see pinclass, cotype)
idhome	char(3)	pin identification only possible at home station, swu, coded (000=no, 001=yes)
transfer	char(3)	transfer of cos and/or key contents from the home station, swu, coded (see pinclass, transfer)
pindur	char(3)	duration of the activated pin, swu, coded (see pinclass, pindur)
pinnkey	char(3)	pin entry on name keys and individual speed facility, swu, coded (000=no, 001=yes)
pintest	char(3)	pin test, swu, coded (see pinclass, pintest)
pintext	char(20)	copin description text, swu
pinlcos2	char(3)	pin input locked if COS2 is active at remote station, swu, coded (000=no, 001=yes)

pinext	char(3)	pin of the forwarding station is retained, swu, coded (000=no, 001=yes)
poschkn	char(2)	position of check number, swu
dptest	char(3)	dptest, swu, coded (000=no, 001=yes)
pintext_e_char	char(20)	pintext_e_char, swu
disptrkc	char(3)	swu, coded (000=no, 001=yes)
pararngm	char(3)	swu, coded (000=no, 001=yes)

10.2.1.78 PINGENERAL

Table 101: PINGENERAL

PINGENERAL		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
pinpoint	char(2)	pin pointer, swu
cosxcos2	char(3)	cosxcos2, swu, coded (000=no, 001=yes)
attempts	char(3)	number of failed attempts to enter the pin, swu
pindisp_1	char(3)	pindisp_1, swu,coded (000=no, 001=yes)
pindisp_2	char(3)	pindisp_2, swu,coded (000=no, 001=yes)
pindisp_3	char(3)	pindisp_3, swu,coded (000=no, 001=yes)
pindisp_4	char(3)	pindisp_4, swu,coded (000=no, 001=yes)
pindisp_5	char(3)	pindisp_5, swu,coded (000=no, 001=yes)
pindisp_6	char(3)	pindisp_6, swu,coded (000=no, 001=yes)
pindisp_7	char(3)	pindisp_7, swu,coded (000=no, 001=yes)
pindisp_8	char(3)	pindisp_8, swu,coded (000=no, 001=yes)
pindisp_9	char(3)	pindisp_9, swu,coded (000=no, 001=yes)
pindisp_10	char(3)	pindisp_10,swu,coded (000=no, 001=yes)
cdrac_1	char(12)	cdrac_1, swu
cdrac_2	char(12)	cdrac_2, swu
cdrac_3	char(12)	cdrac_3, swu

cdrac_4	char(12)	cdrac_4, swu
cdrac_5	char(12)	cdrac_5, swu
cdrac_6	char(12)	cdrac_6, swu
cdrac_7	char(12)	cdrac_7, swu
cdrac_8	char(12)	cdrac_8, swu
cdraclen	char(2)	cdraclen, swu
pindisp_11	char(3)	pindisp_11,swu,coded (000=no, 001=yes)
pindisp_12	char(3)	pindisp_12,swu,coded (000=no, 001=yes)
pindisp_13	char(3)	pindisp_13,swu,coded (000=no, 001=yes)
pindisp_14	char(3)	pindisp_14,swu,coded (000=no, 001=yes)
pindisp_15	char(3)	pindisp_15,swu,coded (000=no, 001=yes)

10.2.1.79 PINPERS (pin/persdat Join)

Table 102: PINPERS

ODF name: Operations:	**PINPERS (pin/persdat Join)** Export: **Select, Select_updates** Import: **-**	
Field	Data type	Description
country	char(2)	country (short code) (P)
company	char(30)	company
org1	char(12)	department
org2	char(12)	department
org3	char(12)	department
name	char(25)	name of the subscriber (P)
sortname	char(30)	name without muted vowels/special characters
christianname	char(15)	first name
title	char(15)	title
salutation	char(15)	address
location	char(5)	location
building	char(5)	building
longname	char(5)	room
extension	char(12)	internal extension
switch	char(4)	Hicom switch

position	int	position
tienum	char(6)	tie line
ext_areacode	char(6)	area code
ext_netcode	char(16)	CO number
ext_ext	char(6)	extension
faxnumber	char(6)	fax number
owner_group	char(30)	ETD user group
direct_trunk_acc	char(1)	flag (for displaying trunk access)
text1	char(24)	key word 1
text2	char(24)	key word 2
text3	char(24)	key word 3
domain	char(8)	domain
pin_num	char(12)	PIN number
pin_type	char(3)	PIN type
vnac	char(6)	
isdn_cc	char(4)	
isdn_ac	char(6)	
isdn_lc	char(10)	
extension_short	char(6)	

10.2.1.80 PORT

Table 103: PORT

PORT		
:EXPORT:SELECT: SELECT_UPDATES		
:IMPORT:INSERT: DELETE:UPDATE		
unique_key	int	unique_key
extension	char(12)	Extension, swu, key
switch_name	char(4)	switch_name, key
pen	char(13)	PEN, swu
devconame	char(20)	Device combination

opstate	char(3)	opstate, swu, coded
chargeid	char(6)	Basic Charge Class
wabegrpnum	char(2)	wabegrpnum, swu
vbzgrpnum	char(2)	vbzgrpnum, swu
tcos	char(4)	Cos 1, swu
ncos	char(4)	Cos 2, swu
berumgrpnum	char(2)	COSX Group, swu
stbnum	char(3)	num_key_lay, swu
bkrnl	char(2)	Speed dial list 1, swu
speed_dial_two	char(2)	Speed dial list 2, swu
speed_dial	char(3)	speed_dial, swu, coded
numdisp	char(3)	displ, swu, coded
devtyp	char(3)	device type, swu, coded
devtext	char(30)	devtext, swu
att_tone_source	char(3)	VC warn tone source, swu, coded
att_tone_target	char(3)	VC warn tone destination, swu, coded
technical_ext	char(12)	Technical extension, swu
switch_new	char(4)	switch_new, swu
temporary	char(3)	Temporary extension, coded
call_back_free	char(3)	Callback-no answer rejection, swu, coded
call_back_busy	char(3)	Callback-busy rejection, swu, coded
displayname	char(31)	Name, swu
dl_idx_man	char(3)	Manual downloading, swu
dl_idx_auto	char(3)	Automatic downloading, swu
domain	char(8)	domain
least_cost_rout1	char(4)	LCRCOS 1, swu
least_cost_rout2	char(4)	LCRCOS 2, swu
least_cost_data1	char(4)	LCRCOSDAT 1, swu
least_cost_data2	char(4)	LCRCOSDAT 2, swu
navar	char(3)	NAVAR, swu, coded
assoc	char(3)	Associated subscriber, swu, coded
analog_modem	char(3)	analogue_modem, swu, coded
clc_num	char(3)	CLC number, swu, coded
bus_type	char(3)	bus_type, swu, coded
bus_extension	char(12)	Main Extension, swu
tsi_voice	char(1)	TSI voice, swu
tsi_data	char(1)	TSI data, swu

tsi_fax	char(1)	TSI fax, swu
protocol_voice	char(3)	Protocol voice, swu, coded
protocol_data	char(3)	Protocol data, swu, coded
protocol_fax	char(3)	Protocol fax, swu, coded
msn	char(3)	MSN, swu, coded
tsi_ttx	char(1)	TSI ttx, swu
tsi_btx	char(1)	TSI vtx, swu
tsi_msn	char(1)	TSI msn, swu
connect_type	char(3)	Connect Type, swu, coded
opt_voice	char(2)	Voice option, swu
opt_data	char(2)	Data option, swu
opt_fax	char(2)	Fax option, swu
secr_device	char(3)	Secretary telephone, swu, coded
info	char(600)	info
fcfw	char(22)	fcfw, swu
compl_mod_allowed	char(3)	compl_mod_allowed, swu, coded
registration	char(3)	Registration, swu, coded
opt_btx	char(2)	VTX option, swu
opt_ttx	char(2)	TTX option, swu
prot_btx	char(3)	VTX protocol, swu, coded
prot_ttx	char(3)	TTX protocol, swu, coded
p_akl	char(3)	KN, swu, coded
p_aufaks	char(3)	RKNOVR, swu, coded
p_aufs	char(3)	KNOVR, swu, coded
p_dirans	char(3)	VC, swu, coded
p_hot	char(3)	HOT, swu, coded
p_natr	char(3)	EOVR, swu, coded
p_roe	char(3)	OFHKRC, swu, coded
p_agent	char(3)	AGENT, swu, coded
p_autolog	char(3)	AUTOLOG, swu, coded
p_caf	char(3)	CAF, swu, coded
p_castmon	char(3)	CASTMON, swu, coded
p_cbstmon	char(3)	CBSTMON, swu, coded
p_dndsr	char(3)	DNDSR, swu, coded
p_super	char(3)	SUPER, swu, coded
p_ani1	char(3)	ANI1, swu, coded
p_ani2	char(3)	ANI2, swu, coded

p_ani3	char(3)	ANI3, swu, coded
p_ani4	char(3)	ANI4, swu, coded
p_ani5	char(3)	ANI5, swu, coded
p_ani6	char(3)	ANI6, swu, coded
p_ani7	char(3)	ANI7, swu, coded
p_ani8	char(3)	ANI8, swu, coded
p_ani9	char(3)	ANI9, swu, coded
p_ani10	char(3)	ANI10, swu, coded
pen_num	int	pen_num, swu
net_config	char(3)	Network-Wide Configuration, coded
log_pty_switch	char(4)	Logical Subscriber System, swu
log_pty_ext	char(12)	Logical Subscriber Number, swu
log_pty_pen	char(13)	Logical Subscriber PEN, swu
log_pty_pen_num	int	log_pty_pen_num, swu
rem_pty_switch	char(4)	Remote Subscriber System, swu
rem_pty_ext	char(12)	Remote Subscriber Number, swu
rem_pty_pen	char(13)	Remote Subscriber PEN, swu
rem_pty_pen_num	int	rem_pty_pen_num, swu
extend_pty_ext	char(12)	Extended Connection Number, swu
ext_pty_ext_l	char(22)	Extension number of extended call, swu
ext_pty_ptnr_ext	char(12)	Destination Data Number
ext_pty_ptnr_ext_l	char(22)	Destination Data Extension Number, swu
ext_pty_ptnr_swi	char(4)	Destination Data System
conf_not_consist	char(3)	conf_not_consist, swu, coded
opt_ntvlv	char(2)	NTVLV Option, swu
prot_ntvlv	char(3)	NTVLV Protocol, swu, coded
permakt	char(3)	permakt, swu, coded
cordless	char(3)	Cordless extern, swu, coded
lwpar	char(2)	lwpar, swu
uart	char(3)	uart, swu, coded
num_callback	char(2)	Max. Number of Callbacks, swu
protocol_spp	char(3)	Protocol S0PP/S2PP, swu, coded
opt_spp	char(2)	Option S0PP/S2PP, swu
alarm	char(4)	Alarm Number, swu
circidx	char(3)	CIRCIDX, swu
cofidx	char(2)	COFIDX, swu
call_free_mail	char(3)	Callback on No Answer, swu, coded

language	char(3)	Preferred Language, swu, coded
p_sprstat	char(3)	Language Static, swu, coded
p_sprdyn	char(3)	Language Dynamic, swu, coded
vpi	char(1)	VPI, swu
vci	char(5)	VCI, swu
status	char(26)	Status, swu
call_log	char(3)	Call Logging, swu, coded
api	char(3)	Work Station Protocol, swu, coded
p_spawh	char(3)	LOADF, swu, coded
p_spsc	char(3)	LOSC, swu, coded
p_spenb	char(3)	LOENB, swu, coded
auto_privacy	char(3)	Automatic Privacy, swu, coded
single_button_mode	char(3)	Single Key Mode, swu, coded
out_line_sel	char(3)	Outgoing Call Preference, swu, coded
inc_line_sel	char(3)	Incoming Call Preference, swu, coded
ring_type	char(3)	Subscriber busy warning tone, swu, coded
key_system	char(3)	Key System, swu, coded
busindex	int	busindex, swu
p_teamchef	char(3)	Executive, swu, coded
p_teamsecr	char(3)	Secretary, swu, coded
p_drziel	char(3)	DSS Dest. networkwide, swu, coded
teamnum	char(3)	Exec./Secr. Team
teamfunct	char(3)	Team Function, swu, coded
store_calls	char(3)	Store calls, swu, coded
dtb_access	char(3)	DTB Access, swu, coded
dual_mode_skey	char(3)	Dual Mode Softkey, swu, coded
sms_feature	char(3)	SMS, swu, coded
reg_counter	char(1)	Registration counter, swu
ic_data_type	char(3)	Data Type, swu, coded
ltu	char(2)	ltu
trace_mal	char(3)	Malicious call trace allowed, swu, coded
asynct	char(4)	asynchronious time, swu
offtype	char(3)	Switching off type, coded
hmusic	char(3)	music on hold recording, swu
extbus	char(3)	Extented bus, swu, coded
pattern	char(3)	Idle code pattern for the B-channels, swu
pulslev	char(3)	Level of charge metering pulses, swu, coded

pulstype	char(3)	Type of charge metering pulses, swu, coded
ackst	char(3)	acknowledge state between dps and announcement device, swu, coded
exact_dev_typ	char(3)	exact device type, swu, coded
exact_dev_text	char(30)	remarks to exact device type, swu
ccmscd	char(6)	multiple code-calling code, swu
ccno	char(6)	code-calling number, swu
spec_outpulse_dev	char(3)	spec_outpulse_dev, swu, coded
special	char(3)	Special station, swu, coded
sufdial	char(3)	Station can suffix dial, swu, coded
witness	char(3)	Witness, swu, coded
phone_mail_idx	char(2)	Index for PhoneMail / Xpression access number, swu
dnidsp	char(3)	display DNIS text, swu, coded
circuit	char(3)	circuit number
flash_enable	char(3)	enable hook flash, swu, coded
ops_conn_type	char(3)	Off Premise connection type, swu, coded
src_grp_idx	char(3)	Source group index, swu
classmrk_ec	char(3)	classmrk_ec, swu, coded
classmrk_g711	char(3)	classmrk_g711, swu, coded
classmrk_g729	char(3)	classmrk_g729, swu, coded
classmrk_vad	char(3)	classmrk_vad, swu, coded
classmrk_init	char(3)	classmrk_init, swu, coded
hotline_idx	char(5)	Hotline Index, swu
node_id	char(10)	node_id, swu
cdracc	char(12)	CDR Account Code, swu
fwd_spk_call	char(3)	fwd_spk_call, swu, coded
p_spk1way	char(3)	p_spk1way, swu, coded
com_group	char(4)	Com Group, swu
com_spk_call_2wy	char(3)	COM group Speaker Call 2-Way, swu, coded
spk_call_1wy_brd	char(3)	spk_call_1wy_brd, swu, coded
p_elapsed_time	char(3)	Elapsed Time, swu, coded
p_block_auto_rel	char(3)	p_block_auto_rel, swu, coded
protocol_class	char(3)	protocol_class, swu, coded
lin1	char(22)	lin1, swu
lin2	char(22)	lin2, swu
acf_appl	char(3)	acf_appl, swu
dtmf_conf	char(3)	dtmf_conf, swu, coded

dhpar_asend	char(3)	dhpar_asend, swu, coded
dhpar_bsend	char(3)	dhpar_bsend, swu, coded
dhpar_dtmfst	char(3)	dhpar_dtmfst, swu, coded
dhpar_flash	char(3)	dhpar_flash, swu, coded
dhpar_vmsdp	char(3)	dhpar_vmsdp, swu, coded
dhpar_ledsign	char(3)	dhpar_ledsign, swu, coded
dhpar_mwled	char(3)	dhpar_mwled, swu, coded
dhpar_predia	char(3)	dhpar_predia, swu, coded
dhpar_ring1	char(3)	dhpar_ring1, swu, coded
bas_svc_audio31k	char(3)	bas_svc_audio31k, swu, coded
bas_svc_voice	char(3)	bas_svc_voice, swu, coded
bas_svc_udi64	char(3)	bas_svc_udi64, swu, coded
bas_svc_udi64ra	char(3)	bas_svc_udi64ra, swu, coded
sztype_atnd	char(3)	sztype_atnd, swu, coded
sztype_dszco	char(3)	sztype_dszco, swu, coded
sztype_dszint	char(3)	sztype_dszint, swu, coded
sztype_expr	char(3)	sztype_expr, swu, coded
sztype_fwd	char(3)	sztype_fwd, swu, coded
sztype_fwdnans	char(3)	sztype_fwdnans, swu, coded
szcd_atnd	char(4)	szcd_atnd, swu
szcd_dszco	char(4)	szcd_dszco, swu
szcd_dszint	char(4)	szcd_dszint, swu
szcd_expr	char(4)	szcd_expr, swu
szcd_fwd	char(4)	szcd_fwd, swu
szcd_fwdnans	char(4)	szcd_fwdnans, swu
pubnum	char(15)	pubnum, swu
ton	char(3)	Ton, swu, coded
npi	char(3)	npi, swu, coded
dtmf_control	char(3)	dtmf_control, swu, coded
delay_cfw_busy	char(3)	delay_cfw_busy, swu, coded
p_cfnr	char(3)	p_cfnr, swu, coded
p_dsfwdext	char(3)	p_dsfwdext, swu, coded
p_mlpp_routine	char(3)	p_mlpp_routine, swu, coded
p_mlpp_prty	char(3)	p_mlpp_prty, swu, coded
p_mlpp_immed	char(3)	p_mlpp_immed, swu, coded
p_mlpp_flash	char(3)	p_mlpp_flash, swu, coded
p_mlpp_flashov	char(3)	p_mlpp_flashov, swu, coded

p_mlpp_prot	char(3)	p_mlpp_prot, swu, coded
p_mlpp_mlpps	char(3)	p_mlpp_mlpps, swu, coded
pen_standard	char(9)	pen_standard, swu
circuit_standard	char(3)	circuit_standard, swu
ippassw	char(20)	ippassw, swu
p_alogontw	char(3)	p_alogontw, swu, coded
classmrk_g729opt	char(3)	classmrk_g729opt, swu, coded
vcepage	char(3)	vcepage, swu, coded
p_reserve_1	char(3)	p_reserve_1, swu, coded
p_reserve_2	char(3)	p_reserve_2, swu, coded
p_reserve_3	char(3)	p_reserve_3, swu, coded
mixed_net_move	char(3)	mixed_net_move, swu, coded
location_code	char(10)	Location Code
extension_limit	char(6)	extension_limit
p_dsswoovr	char(3)	p_dsswoovr, swu, coded
um_operation_id	char(50)	um_operation_id
p_dssline	char(3)	p_dssline, swu, coded
p_rngxferl	char(3)	p_rngxferl, swu, coded
p_srnline	char(3)	p_srnline, swu, coded
p_monslntl	char(3)	p_monslntl, swu, coded
display_mode	char(3)	display_mode, swu, coded
lin_temp	char(22)	lin_temp, swu
emerg_no_temp	char(22)	emerg_no_temp, swu
mob_user_via_app	char(3)	mob_user_via_app, swu, coded
p_mhfatbv	char(3)	p_mhfatbv, swu, coded
p_mhfahome	char(3)	p_mhfahome, swu, coded
stno_oos	char(22)	stno_oos, swu
stno_ap_emrgcy	char(22)	stno_ap_emrgcy, swu
mov_hfa_ip_phone	char(3)	mov_hfa_ip_phone, swu, coded
cluster_id	char(3)	cluster_id, swu
dmcallwd	char(3)	dmcallwd, swu, coded
p_mlpp_rtprot	char(3)	p_mlpp_rtprot, swu, coded
mlpp_svcdom	char(19)	mlpp_svcdom, swu
p_calltype	char(3)	p_calltype, swu, coded
amoaltrt	char(3)	amoaltrt, swu, coded
codec_type	char(3)	codec_type, swu, coded
classmrk_hg3530	char(3)	classmrk_hg3530, swu, coded

classmrk_hg3550	char(3)	classmrk_hg3550, swu, coded
ape_group_no	char(2)	ape_group_no, swu
cmi_handset_type	char(3)	CMI Device Type, coded
ip_address	char(15)	ip_address, swu
rel_status	char(3)	rel_status, swu, coded
retr_counter	int	retr_counter
src_switch	char(4)	src_switch
dest_switch	char(4)	dest_switch
src_pen	char(13)	src_pen
dest_pen	char(13)	dest_pen
cmi_move	char(3)	cmi_move, coded
private_phonebook	char(3)	private_phonebook, coded
auth_class	int	auth_class, swu
simplified_login	char(3)	simplified_login, swu, coded
static_pin	char(8)	static_pin, swu
p_dsslinesign	char(3)	swu, coded
p_normalprio	char(3)	swu, coded
p_highprio	char(3)	swu, coded
line_tone	char(3)	swu, coded
busy_led_sign	char(3)	swu, coded
same_prio	char(3)	swu, coded
prio_call	char(3)	swu, coded
sip_ip_address	char(15)	swu
sip_secret	char(19)	swu
sip_userid	char(19)	swu
sip_realm	char(19)	swu
sip_pre_auth_user	char(3)	swu, coded
sip_auth_required	char(3)	swu, coded
sip_smg_active	char(3)	swu, coded
sip_s2_mld	char(3)	swu
tclass	char(2)	swu
reloc_time	int	
list_id	int	
reg_type	char(3)	coded
tenant_group	char(20)	Note: **Manager** only
pin_num	char(12)	
pin_type	char(3)	coded

p_nodiv	char(3)	swu, coded
save_assoc_obj	char(3)	coded
dmc_secured	char(3)	swu, coded
para_rng_master	char(3)	swu, coded
para_rng_subscr	char(3)	swu, coded
appm	char(3)	swu, coded
long_repr	char(3)	coded
vnac	char(6)	
dls_name	char(20)	
mac_address	char(17)	
dls_standard_name	char(40)	
ip_device_type	char(50)	
loadware_variant	char(30)	
dls_support	char(3)	coded
mark_for_move	char(19)	
ret_htg_mbrshp	char(3)	coded
ret_pug_mbrshp	char(3)	coded
other_mbrshp_lost	char(3)	coded
p_disspkmc	char(3)	swu, coded
cord_e_pin_exists	char(3)	coded
extension_short	char(6)	
p_disppup	char(3)	swu, coded
p_dispfwd	char(3)	swu, coded
p_dispches	char(3)	swu, coded
security_class	char(3)	swu, coded
sec_level_tone	char(3)	swu, coded
mbxdspid	char(3)	swu, coded
seclvdsp	char(3)	swu, coded
neonmwi	char(3)	swu, coded
notrfpln	char(3)	swu, coded
twin_num	char(12)	swu
blf	char(3)	swu, coded
ltu_location	char(48)	
cmi_handset_string	char(20)	swu
cordless_bulk_add	char(3)	coded
manuel_bulk_add	char(3)	coded
p_anate_call_wait	char(3)	swu, coded

p_secure_only_sip	char(3)	swu, coded
addon_switch	char(3)	coded
bulkadd_first_pen	char(13)	
p_noctimic	char(3)	swu, coded
src_node_id	char(13)	
dest_node_id	char(13)	
template_id	char(12)	
ufa_type	char(3)	swu, coded
ufa_stno	char(12)	swu
no_call_fwd_ign	char(3)	swu, coded
cust_lan_ip_addr	char(15)	swu
port_tcp	char(5)	swu
port_tls	char(5)	swu
pubname	char(22)	swu
e164_num	char(22)	swu
email_addr	char(127)	swu
assdev_type	char(1)	swu
assdev_prim	char(12)	swu
assdev_clnt	char(12)	swu
assdev_sec1	char(12)	swu
assdev_sec2	char(12)	swu

10.2.1.81 PRIV_DIAL_NUMBER

Table 104: PRIV_DIAL_NUMBER

PRIV_DIAL_NUMBER		
:EXPORT:SELECT		
:IMPORT:INSERT, UPDATE,DELETE		
unique_key	int	unique_key
switch_name	char(4)	System, key
domain	char(8)	domain, swu
location_code	char(6)	Location Code, swu
station_no	char(6)	Station No., swu, key
station_no_limit	char(6)	Station No. Limit, swu

station_no_2	char(6)	station_no_2, swu, key
private_number	char(12)	Private Number, swu
priv_no_limit	char(12)	Private Number Limit, swu
type	char(3)	type, swu, key, coded (see Switches 100-149, Switches 150-179),
status	char(3)	Status, swu, coded (see General Coded Values (0-10), 002=free, 003=used)
reserved	char(3)	Reserved, coded (000=no, 001=yes)
res_sb_name	char(10)	Reserved for sched. Batch
reserved_pen	char(15)	For PEN
reserved_pen_num	int	reserved_pen_num
reserved_time	int	Reservation expires at
cons_flag	char(3)	Consistent in Domain, swu, coded(000=no, 001=yes)
cons_id	int	cons_id, swu
loc_flag	char(1)	loc_flag, swu
subdomain	char(8)	subdomain
res_without_pen	char(3)	res_without_pen, coded 000/001
last_inconst_check	char(10)	last inconsistent check
rsrvd_switch_name	char(4)	
tenant_group	char(20)	Note: **Manager** only

10.2.1.82 PRIV_DIAL_NUM_RA

Table 105: PRIV_DIAL_NUM_RA

PRIV_DIAL_NUM_RA		private dial number range
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	System, key
domain	char(8)	domain, swu
dms_type	char(3)	dms_type, key, coded
station_no_range	char(6)	station_no_range, swu, key
info	char(300)	info

10.2.1.83 PRIV_DIAL_NUM_ST

Table 106: PRIV_DIAL_NUM_ST

PRIV_DIAL_NUM_ST		private dial number range
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	System, key
domain	char(8)	domain, swu
station_no	char(6)	
virtual_node_id	char(12)	
status	char(3)	

10.2.1.84 PRIV_DIAL_NUM_WV

Table 107: PRIV_DIAL_NUM_WV

PRIV_DIAL_NUMBER		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	System, key
domain	char(8)	domain, swu
station_no	char(6)	Station No., swu, key
type	char(3)	type, swu, key, coded (see Switches 100-149, Switches 150-179),
status	char(3)	Status, swu, coded (see General Coded Values (0-10), 002=free, 003=used)
dpln	char(2)	dpln, swu, key
cps0	char(3)	cps0, swu, coded
...		
cps22	char(3)	cps22, swu, coded
wabe0	char(3)	swu, coded
...		
wabe15	char(3)	swu, coded
alldpln	char(37)	

10.2.1.85 ROUTE_INFO_WN

Table 108: ROUTE_INFO_WN

ROUTE_INFO_WN		Route info
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	System, key
domain	char(8)	domain, swu
station_no	char(6)	Station No., swu, key
type	char(3)	type, swu, coded
route_index	char(4)	route_index, swu
dnno	char(10)	dnno, swu
process_flag	char(3)	process_flag, swu, coded
dpln	char(2)	dpln, swu, key
cps	char(2)	Call Process State, swu, key
arn	char(2)	Alternate Reference Number, swu
cfremvar	char(3)	Cf-Remote-Varienat, swu, coded
cfremse	char(3)	Cf-Remote-Service, swu, coded

10.2.1.86 ROUTE_INFO_WT

Table 109: ROUTE_INFO_WT

ROUTE_INFO_WT		Route info subscriber
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	System, key
domain	char(8)	domain, swu
station_no	char(12)	Station No., swu, key
route_index	char(4)	route_index, swu
dnno	char(10)	dnno, swu
process_flag	char(3)	process_flag, swu, coded
cps_2	char(3)	cps_2, swu, coded
cps_3	char(3)	cps_3, swu, coded

cps_4	char(3)	cps_4, swu, coded
cps_5	char(3)	cps_5, swu, coded
cps_6	char(3)	cps_6, swu, coded
cps_7	char(3)	cps_7, swu, coded
cps_8	char(3)	cps_8, swu, coded
cps_9	char(3)	cps_9, swu, coded
cps_10	char(3)	cps_10, swu, coded
cps_11	char(3)	cps_11, swu, coded
cps_12	char(3)	cps_12, swu, coded
cps_22	char(3)	cps_22, swu, coded
dpln	char(2)	swu, key
station_no_short	char(6)	
long_repr	char(3)	coded

10.2.1.87 SBS_PORT

Table 110: SBS_PORT

SBS_PORT		
:EXPORT:SELECT:SELECT_UPDATES		
:IMPORT:INSERT:DELETE:UPDATE		
unique_key	int	unique_key
extension	char(12)	extension, swu, key
switch_name	char(4)	switch_name, key
pen	char(13)	pen, swu
devconame	char(20)	Device combination
opstate	char(3)	swu, coded
chargeid	char(6)	
wabegrpnum	char(2)	swu
vbzgrpnum	char(2)	swu
tcos	char(4)	Cos 1, swu
ncos	char(4)	Cos 2, swu
berumgrpnum	char(2)	swu

stbnum	char(3)	num_key_lay, swu
bkrnl	char(2)	Speed dial list 1, swu
speed_dial_two	char(2)	Speed dial list 2, swu
speed_dial	char(3)	swu, coded
numdisp	char(3)	displ, swu, coded
devtyp	char(3)	device type, swu, coded
devtext	char(15)	swu
att_tone_source	char(3)	VC warn tone source, swu, coded
att_tone_target	char(3)	VC warn tone destination, swu, coded
technical_ext	char(6)	Technical extension, swu
switch_new	char(4)	swu
temporary	char(3)	Temporary extension, coded
call_back_free	char(3)	Callback-no answer rejection, swu, coded
call_back_busy	char(3)	Callback-busy rejection, swu, coded
name	char(31)	swu
dl_idx_man	char(3)	Manual downloading, swu
dl_idx_auto	char(3)	Automatic downloading, swu
domain	char(8)	domain
least_cost_rout1	char(2)	LCRCOS 1, swu
least_cost_rout2	char(2)	LCRCOS 2, swu
least_cost_data1	char(2)	LCRCOSDAT 1, swu
least_cost_data2	char(2)	LCRCOSDAT 2, swu
navar	char(3)	swu, coded
assoc	char(3)	Associated subscriber, swu, coded
analog_modem	char(3)	swu, coded
clc_num	char(3)	CLC number, swu, coded
bus_type	char(3)	swu, coded
bus_extension	char(6)	swu
tsi_voice	char(1)	TSI voice, swu
tsi_data	char(1)	TSI data, swu
tsi_fax	char(1)	TSI fax, swu
protocol_voice	char(3)	Protocol voice, swu, coded

protocol_data	char(3)	Protocol data, swu, coded
protocol_fax	char(3)	Protocol fax, swu, coded
msn	char(3)	swu, coded
tsi_ttx	char(1)	TSI ttx, swu
tsi_btx	char(1)	TSI vtx, swu
tsi_msn	char(1)	TSI msn, swu
connect_type	char(3)	swu, coded
opt_voice	char(2)	Voice option, swu
opt_data	char(2)	Data option, swu
opt_fax	char(2)	Fax option, swu
secr_device	char(3)	Secretary telephone, swu, coded
info	char(50)	info
fcfw	char(22)	swu
compl_mod_allowed	char(3)	swu, coded
registration	char(3)	Registration, swu, coded
opt_btx	char(2)	VTX option, swu
opt_ttx	char(2)	TTX option, swu
prot_btx	char(3)	VTX protocol, swu, coded
prot_ttx	char(3)	TTX protocol, swu, coded
p_akl	char(3)	KN, swu, coded
p_aufaks	char(3)	RKNOVR, swu, coded
p_aufs	char(3)	KNOVR, swu, coded
p_dirans	char(3)	VC, swu, coded
p_hot	char(3)	HOT, swu, coded
p_natr	char(3)	EOVR, swu, coded
p_roe	char(3)	OFHKRC, swu, coded
p_agent	char(3)	AGENT, swu, coded
p_autolog	char(3)	AUTOLOG, swu, coded
p_caf	char(3)	CAF, swu, coded
p_castmon	char(3)	CASTMON, swu, coded
p_cbstmon	char(3)	CBSTMON, swu, coded
p_dndsr	char(3)	DNDSR, swu, coded

p_super	char(3)	SUPER, swu, coded
p_ani1	char(3)	ANI1, swu, coded
p_ani2	char(3)	ANI2, swu, coded
p_ani3	char(3)	ANI3, swu, coded
p_ani4	char(3)	ANI4, swu, coded
p_ani5	char(3)	ANI5, swu, coded
p_ani6	char(3)	ANI6, swu, coded
p_ani7	char(3)	ANI7, swu, coded
p_ani8	char(3)	ANI8, swu, coded
p_ani9	char(3)	ANI9, swu, coded
p_ani10	char(3)	ANI10, swu, coded
pen_num	int	swu
net_config	char(3)	Network-Wide Configuration, coded
log_pty_switch	char(4)	Logical Subscriber System, swu
log_pty_ext	char(6)	Logical Subscriber Number, swu
log_pty_pen	char(13)	Logical Subscriber PEN, swu
log_pty_pen_num	int	swu
rem_pty_switch	char(4)	Remote Subscriber System, swu
rem_pty_ext	char(6)	Remote Subscriber Number, swu
rem_pty_pen	char(13)	Remote Subscriber PEN, swu
rem_pty_pen_num	int	swu
extend_pty_ext	char(6)	Extended Connection Number, swu
ext_pty_ext_l	char(22)	Extension number of extended call, swu
ext_pty_ptnr_ext	char(6)	Destination Data Number
ext_pty_ptnr_ext_l	char(22)	Destination Data Extension Number, swu
ext_pty_ptnr_swi	char(4)	Destination Data System
conf_not_consist	char(3)	swu, coded
opt_ntvlv	char(2)	NTVLV Option, swu
prot_ntvlv	char(3)	NTVLV Protocol, swu, coded
permakt	char(3)	swu, coded
cordless	char(3)	swu, coded
lwpar	char(2)	swu

uart	char(3)	swu, coded
num_callback	char(2)	Max. Number of Callbacks, swu
protocol_spp	char(3)	Protocol S0PP/S2PP, swu, coded
opt_spp	char(2)	Option S0PP/S2PP, swu
alarm	char(4)	Alarm Number, swu
circidx	char(3)	CIRCIDX, swu
cofidx	char(2)	COFIDX, swu
call_free_mail	char(3)	Callback on No Answer, swu, coded
language	char(3)	Preferred Language, swu, coded
p_sprstat	char(3)	Language Static, swu, coded
p_sprdyn	char(3)	Language Dynamic, swu, coded
vpi	char(1)	VPI, swu
vci	char(5)	VCI, swu
status	char(26)	Status, swu
call_log	char(3)	Call Logging, swu, coded
api	char(3)	Work Station Protocol, swu, coded
p_spawh	char(3)	LOADF, swu, coded
p_spsc	char(3)	LOSC, swu, coded
p_spenb	char(3)	LOENB, swu, coded
auto_privacy	char(3)	Automatic Privacy, swu, coded
single_button_mode	char(3)	Single Key Mode, swu, coded
out_line_sel	char(3)	Outgoing Call Preference, swu, coded
inc_line_sel	char(3)	Incoming Call Preference, swu, coded
ring_type	char(3)	Subscriber busy warning tone, swu, coded
key_system	char(3)	Key System, swu, coded
busindex	int	swu
p_teamchef	char(3)	Executive, swu, coded
p_teamsecr	char(3)	Secretary, swu, coded
p_drziel	char(3)	DSS Dest. networkwide, swu, coded
teamnum	char(3)	Exec./Secr. Team, swu
teamfunct	char(3)	Team Function, swu, coded
store_calls	char(3)	Store calls, swu, coded

dtb_access	char(3)	DTB Access, swu, coded
dual_mode_skey	char(3)	Dual Mode Softkey, swu, coded
sms_feature	char(3)	SMS, swu, coded
reg_counter	char(1)	Registration counter, swu
ic_data_type	char(3)	Data Type, swu, coded
ltu	char(2)	ltu
trace_mal	char(3)	malicious call trace allowed, swu, coded
asynct	char(4)	asynchronious time, swu
offtype	char(3)	switching off type, swu, coded
hmusic	char(3)	music on hold recording, swu
extbus	char(3)	extented bus, swu, coded
pattern	char(3)	idlecode pattern for the b-channels, swu
pulslev	char(3)	level of charge metering pulses, swu, coded
pulstype	char(3)	type of charge metering pulses, swu, coded
ackst	char(3)	acknowledge state between dps and announcement device, swu, coded
exact_dev_typ	char(3)	exact device type, swu, coded
exact_dev_text	char(15)	remarks to exact device type, swu
ccmscd	char(6)	multiple code-calling code, swu
ccno	char(6)	code-calling number, swu
spec_outpulse_dev	char(3)	spec_outpulse_dev, swu, coded
special	char(3)	special station, swu, coded
sufdial	char(3)	station can suffix dial, swu, coded
witness	char(3)	witness, swu, coded
phone_mail_idx	char(2)	index for PhoneMail / Xpression access number, swu
dnidsp	char(3)	display DNIS text, swu, coded
circuit	char(3)	circuit number, coded
flash_enable	char(3)	enable hook flash, swu, coded
ops_conn_type	char(3)	Off Premise connection type, swu, coded
src_grp_idx	char(2)	swu
classmrk_ec	char(3)	swu, coded
classmrk_g711	char(3)	swu, coded

classmrk_g729	char(3)	swu, coded
classmrk_vad	char(3)	swu, coded
classmrk_init	char(3)	swu, coded
hotline_idx	char(3)	Hotline Index, swu
node_id	char(10)	swu
cdracc	char(12)	swu
fwd_spk_call	char(3)	swu, coded
p_spk1way	char(3)	swu, coded
com_group	char(4)	swu
com_spk_call_2wy	char(3)	swu, coded
spk_call_1wy_brd	char(3)	swu, coded
p_elapsed_time	char(3)	swu, coded
p_block_auto_rel	char(3)	swu, coded
protocol_class	char(3)	swu, coded
lin1	char(22)	swu
lin2	char(22)	swu
acf_appl	char(3)	swu
dtmf_conf	char(3)	swu, coded
dhpar_asend	char(3)	swu, coded
dhpar_bsend	char(3)	swu, coded
dhpar_dtmfst	char(3)	swu, coded
dhpar_flash	char(3)	swu, coded
dhpar_vmsdp	char(3)	swu, coded
dhpar_ledsign	char(3)	swu, coded
dhpar_mwled	char(3)	swu, coded
dhpar_predia	char(3)	swu, coded
dhpar_ring1	char(3)	swu, coded
bas_svc_audio31k	char(3)	swu, coded
bas_svc_voice	char(3)	swu, coded
bas_svc_udi64	char(3)	swu, coded
bas_svc_udi64ra	char(3)	swu, coded
sztype_atnd	char(3)	swu, coded

sztype_dszco	char(3)	swu, coded
sztype_dszint	char(3)	swu, coded
sztype_expr	char(3)	swu, coded
sztype_fwd	char(3)	swu, coded
sztype_fwdnans	char(3)	swu, coded
szcd_atnd	char(4)	swu
szcd_dszco	char(4)	swu
szcd_dszint	char(4)	swu
szcd_expr	char(4)	swu
szcd_fwd	char(4)	swu
szcd_fwdnans	char(4)	swu
pubnum	char(15)	swu
ton	char(3)	swu, coded
npi	char(3)	swu, coded
dtmf_control	char(3)	swu, coded
delay_cfw_busy	char(3)	swu, coded
p_cfnr	char(3)	swu, coded
p_dsfwdext	char(3)	swu, coded
p_mlpp_routine	char(3)	swu, coded
p_mlpp_prty	char(3)	swu, coded
p_mlpp_immed	char(3)	swu, coded
p_mlpp_flash	char(3)	swu, coded
p_mlpp_flashov	char(3)	swu, coded
p_mlpp_prot	char(3)	swu, coded
p_mlpp_mlpps	char(3)	swu, coded
pen_standard	char(9)	
circuit_standard	char(3)	
ippassw	char(20)	
p_alogontw	char(3)	
classmrk_g729opt	char(3)	
vcepage	char(3)	
p_reserve_1	char(3)	

p_reserve_2	char(3)
p_reserve_3	char(3)
mixed_net_move	char(3)
location_code	char(10)
extension_limit	char(6)
p_dsswoovr	char(3)
um_operation_id	char(50)
p_dssline	char(3)
p_rngxferl	char(3)
p_srnline	char(3)
p_monslntl	char(3)
display_mode	char(3)
lin_temp	char(22)
emerg_no_temp	char(22)
mob_user_via_app	char(3)
p_mhfatbv	char(3)
p_mhfahome	char(3)
stno_oos	char(22)
stno_ap_emrgcy	char(22)
mov_hfa_ip_phone	char(3)
dmcallwd	char(3)
cluster_id	char(3)
codec_type	char(3)

10.2.1.88 SHELF_DATA

Table 111: SHELF_DATA

SHELF_DATA		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain

ltu	char(2)	ltu, key
shelf_type	char(3)	LTU type, swu, coded (see Flex-Shelf-Values)
description	char(48)	info, swu
phone_no	char(22)	phone number at shelf location, swu
fax_no	char(22)	fax number at shelf location, swu

10.2.1.89 SPEC_SUBSCRIBER

Table 112: SPEC_SUBSCRIBER

SPEC_SUBSCRIBER		special subscriber / Sonderteilnehmer
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
spec_sub_type	char(3)	spec_sub_type, swu, coded
spec_extension	char(3)	spec_extension, swu
pen	char(13)	PEN, swu, key
pen_num	int	pen_num, swu
ltu	char(2)	ltu
info	char(300)	info
line_dist	char(3)	line_dist, swu, coded
in_service	char(3)	in_service, swu, coded
itr_group	char(2)	internal traffic restriction group, swu
cos	char(4)	cos, swu
lcos1	char(4)	lcos1, swu
dpln	char(2)	dpln, swu
cp_alert	char(3)	Notification/Alerting tone, swu, coded
cp_annicpt	char(3)	Announcement for attendant intercept, swu, coded
cp_bonhk	char(3)	B party on hook, swu, coded
cp_busy	char(3)	Busy tone, swu, coded
cp_camp	char(3)	Knocking/Camp-on/Call waiting tone, swu, coded
cp_cbkalert	char(3)	Callback alerting tone, swu, coded

cp_ccmabsen	char(3)	CCM tracer in rack, swu, coded
cp_cmaltone	char(3)	Call monitoring alerting tone, swu, coded
cp_conftn	char(3)	Conference tone, swu, coded
cp_cong	char(3)	Congestion tone, swu, coded
cp_data	char(3)	Data tone, swu, coded
cp_disctone	char(3)	Disconnect tone, swu, coded
cp_dnd	char(3)	Do-not-disturb tone, swu, coded
cp_extdtn	char(3)	External dial tone, swu, coded
cp_extdtn2	char(3)	Secondary CO dial tone, swu, coded
cp_fcamp	char(3)	First camp-on tone, swu, coded
cp_frngbk	char(3)	First burst of ringback tone, swu, coded
cp_hold	char(3)	Call waiting tone, swu, coded
cp_holdccs	char(3)	Hold tone code calling system, swu, coded
cp_holdext	char(3)	Hold tone external, swu, coded
cp_holdh1	char(3)	Hold tone hold 1, swu, coded
cp_holdh2	char(3)	Hold tone hold 2, swu, coded
cp_holdhunt	char(3)	Hold tone hunting-group, swu, coded
cp_holdint	char(3)	Hold tone internal, swu, coded
cp_holdline	char(3)	Hold the line please, swu, coded
cp_huntgrup	char(3)	Announcement hunt group queue, swu, coded
cp_impossfm	char(3)	Impossible for the moment, swu, coded
cp_intdtn	char(3)	Internal dial tone, swu, coded
cp_knconf	char(3)	Camp-on tone for conf, swu, coded
cp_locann	char(3)	Local announcement, swu, coded
cp_music	char(3)	Music on hold, swu, coded
cp_mwi	char(3)	Message waiting information, swu, coded
cp_nack	char(3)	Negative acknowledgement tone, swu, coded
cp_nodial	char(3)	Timeout tone if digits not dialled, swu, coded
cp_nopo	char(3)	Not possible, swu, coded
cp_notalwd	char(3)	Not allowed, swu, coded
cp_notreach	char(3)	Not reachable, swu, coded
cp_nu	char(3)	Number obtainable tone, swu, coded

cp_ovr	char(3)	Automatic override tone, swu, coded
cp_ovrtn	char(3)	Override tone, swu, coded
cp_ovrwarn	char(3)	Override warning tone, swu, coded
cp_pack	char(3)	Positive acknowledgement tone, swu, coded
cp_packwd	char(3)	Positive acknowledgement tone with dialing, swu, coded
cp_postdial	char(3)	Announcement post dial, swu, coded
cp_reorder	char(3)	Reorder tone, swu, coded
cp_rngbk	char(3)	Ringback tone, swu, coded
cp_rto	char(3)	RTO test tone, swu, coded
cp_spec	char(3)	Special dial tone, swu, coded
cp_test	char(3)	Test tone, swu, coded
cp_voicemal	char(3)	VoiceMail, swu, coded
classmrk_ec	char(3)	swu, coded
classmrk_g711	char(3)	swu, coded
classmrk_g729	char(3)	swu, coded
classmrk_g729opt	char(3)	swu, coded
classmrk_vad	char(3)	swu, coded
classmrk_ml	char(3)	classmrk_ml, swu, coded
classmrk_viahhs	char(3)	classmrk_viahhs, swu, coded
ltu_assign	char(4)	
ltu_min	char(2)	
ltu_max	char(2)	
cp_recdialt	char(3)	swu, coded

10.2.1.90 STATION_RANGE

Table 113: STATION_RANGE

STATION_RANGE		
:EXPORT:SELECT		
:IMPORT:INSERT:DELETE:UPDATE		
unique_key	int	unique_key
station_from	char(6)	

station_to	char(6)
switch_name	char(4)
domain	char(8)

10.2.1.91 STATION_XREF

Table 114: STATION_XREF

STATION_XREF		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
extension	char(12)	extension, key
key_number	char(2)	number of the key at the phone, key
key_func	char(3)	function of the key, coded (see key functions (850-899), key functions (920-949))
destination	char(22)	destination
xref_type	char(3)	xref_type, key, coded (see station_xref, xref_type)

10.2.1.92 SWITCH

Table 115: SWITCH

SWITCH		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
switch_typ	char(15)	System type
version	char(5)	version
info	char(300)	info
country	char(2)	country
areacode	char(16)	id
netcode	char(16)	ext_areacode
upload	char(1)	upload

amo_lang	char(3)	amo_lang, coded (006=german, 007=english)
sys_verify	int	sys_verify
nodenum	char(10)	nodenum
version_old	char(5)	version_old
last_delta	char(19)	Last upload delta
last_all	char(19)	Last upload all
domain	char(8)	domain
upload_cnt	int	Upload counter
upload_day	char(3)	PBX data, coded (see Time Values)
exp_upl_date	char(19)	Next delta
spdl_acc_cod1	char(6)	Access code for speed dial list 1
spdl_acc_cod2	char(6)	Access code for speed dial list 2
admin	char(3)	PBX data, coded (000=no, 001=yes)
vms_admin	char(3)	VMS data, coded (000=no, 001=yes)
vms_server	char(3)	VM server, coded (see Server 537-549)
vms_switch	char(4)	VMS switch
vms_last_delta	char(19)	Last VMS delta
vms_last_all	char(19)	Last VMS all
vms_sys_verify	int	vms_sys_verify
vms_upload	char(1)	VMS upload
vms_upload_cnt	int	VMS upload counter
vms_dbsync	char(1)	vms_dbsync
switch_dbsync	char(1)	switch_dbsync
cmi_button_std	char(3)	CMI key layout
cmi_cos	char(4)	CMI cos
cmi_lcos	char(4)	CMI-LCOS
lcr_data	char(3)	LCR support, coded (000=no, 001=yes)
lcr_last_all	char(19)	Last LCR upload all
lcr_last_delta	char(19)	Last LCR upload delta
lcr_upload_cnt	int	Upload counter for LCR
lcr_upload_day	char(3)	LCR data, coded
lcr_dbsync	char(1)	Upload flag for LCR

lcr_upload	char(1)	Upload status
lcr_admin	char(3)	LCR data, coded (000=no, 001=yes)
delta_ctrl	char(3)	Request upload data via the DMS, coded (000=no, 001=yes)
sec_node	char(10)	Extended node number
language1	char(3)	Language 1, coded (see Languages)
language2	char(3)	Language 2, coded (see Languages)
language3	char(3)	Language 3, coded (see Languages)
language4	char(3)	Language 4, coded (see Languages)
language5	char(3)	Language 5, coded (see Languages)
countrycode	char(4)	Country Table
uplo2_aps	char(13)	UPLO2 APS
uplol_aps	char(13)	UPLOL APS
uplvm_aps	char(13)	UPLVM APS
knotenkz	char(22)	Private code
consistent	char(3)	consistent
net_team	char(3)	Exec./Secr. Team Administration, coded (000=no, 001=yes)
dtb_server	char(20)	DTB_Server
upload_txt	char(3)	upload status, coded (see switch, upload_txt, Error states (for action protocol))
vms_upload_txt	char(3)	vms upload status, coded (see switch, upload_txt, Error states (for action protocol))
lcr_upload_txt	char(3)	lcr upload status, coded (see switch, upload_txt, Error states (for action protocol))
switch_number	char(17)	switch_number
manual_plen	char(2)	Manual PIN Length
passwd_plen	char(2)	Authorization PIN Length
pref_route_index	char(4)	Preferred Route Index
preliminary	char(3)	Preliminary, coded
preliminary_info	int	preliminary_info
account_length	char(2)	account_length
dss_partners	char(2)	dss_partners
sysm_flag	char(3)	sysm_flag, coded

no_name	char(16)	no_name
pbx_upload	char(1)	pbx_upload
not_upload	char(1)	not_upload
not_dbsync	char(1)	not_dbsync
not_last_all	char(19)	not_last_all
backup_swu	char(1)	backup_swu
upload_file	char(1)	upload_file
upload_him_swu	char(1)	upload_him_swu
upload_him_adp	char(1)	upload_him_adp
him_dbsync	char(1)	him_dbsync
him_last_all	char(19)	him_last_all
startupd_time	char(5)	startupd_time
delta_time	char(3)	delta_time
pbx_upload_txt	char(3)	coded
not_upload_txt	char(3)	coded
upload_file_txt	char(3)	coded
upl_him_swu_txt	char(3)	coded
upl_him_adp_txt	char(3)	coded
hicom_sw_version	char(13)	hicom_sw_version
epnp_barrier	char(6)	epnp_barrier
subdomain	char(50)	subdomain
transit_switch	char(3)	transit_switch
transit_index	char(4)	transit_index
lcr_exp_upl_date	char(19)	lcr_exp_upl_date
legk_support	char(3)	legk_support, coded
dbsync_lock	char(3)	dbsync lock, coded
cmi_max_subsc	char(5)	cmi_max_ subsc
pbx_upl_hour_adp	char(2)	
pbx_upl_min_adp	char(2)	
lcr_upl_hour_adp	char(2)	
lcr_upl_min_adp	char(2)	
pbx_upl_hour_mgr	char(2)	

pbx_upl_min_mgr	char(2)	
lcr_upl_hour_mgr	char(2)	
lcr_upl_min_mgr	char(2)	
tenant_group	char(20)	Note: **Manager** only
cmi_cos2	char(4)	
cmi_lcos2	char(4)	
cmi_lcosd	char(4)	
cmi_lcosd2	char(4)	
wabegrpnum	char(2)	
vbzgrpnum	char(2)	
nunp_active	char(3)	coded
spe_support	char(3)	swu, coded
sec_tdm_subsr	char(3)	swu, coded
sec_tdm_trunk	char(3)	swu, coded
sec_ip_subsr	char(3)	swu, coded
sec_ip_trunk	char(3)	swu, coded
dns_ip_address	char(15)	swu
dls_ip_address	char(15)	swu
dls_port	char(5)	swu
key_len	char(3)	swu, coded
mikey_opt	char(3)	swu, coded
dls_accept_pass	char(3)	swu, coded
vnrad_active	char(3)	coded
vnr_switch_name	char(4)	
seclvdsp	char(3)	swu, coded
upl_time_err	char(3)	coded
display_mode	char(3)	swu, coded
dns_ip_addres2	char(15)	swu
version_string	char(10)	

10.2.1.93 TRANS_SERVICE

Table 116: TRANS_SERVICE

TRANS_SERVICE		
:EXPORT:TRANSLATE		
unique_key	int	unique_key
input_fqn	char(32)	
dialed_digits	char(32)	
fully_qual_num	char(32)	
operator_id	char(32)	
result	int	
result_code	int	

10.2.1.94 UC_ACTION_LOG

Table 117: UC_ACTION_LOG

UC_ACTION_LOG		
:EXPORT:SELECT		
unique_key	int	unique_key
operation	char(20)	
dmsapi_type	char(20)	
time_create_str	char(22)	
switch_name_src	char(4)	
switch_name_dest	char(4)	
station_no_src	char(12)	
station_no_dest	char(12)	
kndef_src	char(10)	
kndef_dest	char(10)	
dialplan_src	char(2)	
dialplan_dest	char(2)	
display_name_src	char(32)	
display_name_dest	char(32)	
dpln_src	char(2)	

dpln_dest	char(2)
npi_src	char(10)
npi_dest	char(10)
ton_src	char(10)
ton_dest	char(10)
exit_src	char(6)
exit_dest	char(6)
prefix_src	char(4)
prefix_dest	char(4)
isdn_cc_src	char(4)
isdn_cc_dest	char(4)
isdn_ac_src	char(7)
isdn_ac_dest	char(7)
isdn_lc_src	char(10)
isdn_lc_dest	char(10)
isdn_skip_dg_src	char(2)
isdn_skip_dg_dest	char(2)
pnp_l2_src	char(4)
pnp_l2_dest	char(4)
pnp_l1_src	char(6)
pnp_l1_dest	char(6)
pnp_l0_src	char(10)
pnp_l0_dest	char(10)
pnp_skip_dg_src	char(2)
pnp_skip_dg_dest	char(2)
virtual_node_src	char(6)
virtual_node_dest	char(6)
skip_dg_src	char(1)
skip_dg_dest	char(1)
vnode_id	char(10)
isVNR_src	char(1)
isVNR_dest	char(1)

pd_src	char(6)
pd_dest	char(6)
pin_num_src	char(12)
pin_num_dest	char(12)
pin_type_src	char(3)
pin_type_dest	char(3)
pin_class_src	char(2)
pin_class_dest	char(2)

10.2.1.95 UPD_PERSDAT

Table 118: UPD_PERSDAT

UPD_PERSDAT		
:EXPORT:SELECT		
unique_key	int	unique_key
delete_time	char(17)	date
xpr_server	char(25)	XPR Server
xpr_key	char(25)	Mailbox ID

10.2.1.96 UPD_PIN

Table 119: UPD_PIN

UPD_PIN		
:EXPORT:SELECT		
unique_key	int	unique_key
delete_time	char(17)	date

10.2.1.97 UPD_PORT

Table 120: UPD_PORT

UPD_PORT		
:EXPORT:SELECT		
unique_key	int	unique_key, key

delete_time	char(17)	date

10.2.1.98 VBZ

Table 121: VBZ

VBZ		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
src00_dst00	char(3)	src00_dst00, swu, coded (000=no, 001=yes)
src00_dst01	char(3)	src00_dst01, swu, coded (000=no, 001=yes)
src00_dst02	char(3)	src00_dst02, swu, coded (000=no, 001=yes)
src00_dst03	char(3)	src00_dst03, swu, coded (000=no, 001=yes)
src00_dst04	char(3)	src00_dst04, swu, coded (000=no, 001=yes)
src00_dst05	char(3)	src00_dst05, swu, coded (000=no, 001=yes)
src00_dst06	char(3)	src00_dst06, swu, coded (000=no, 001=yes)
src00_dst07	char(3)	src00_dst07, swu, coded (000=no, 001=yes)
src00_dst08	char(3)	src00_dst08, swu, coded (000=no, 001=yes)
src00_dst09	char(3)	src00_dst09, swu, coded (000=no, 001=yes)
src00_dst10	char(3)	src00_dst10, swu, coded (000=no, 001=yes)
src00_dst11	char(3)	src00_dst11, swu, coded (000=no, 001=yes)
src00_dst12	char(3)	src00_dst12, swu, coded (000=no, 001=yes)
src00_dst13	char(3)	src00_dst13, swu, coded (000=no, 001=yes)
src00_dst14	char(3)	src00_dst14, swu, coded (000=no, 001=yes)
src00_dst15	char(3)	src00_dst15, swu, coded (000=no, 001=yes)
src01_dst00	char(3)	src01_dst00, swu, coded (000=no, 001=yes)
src01_dst01	char(3)	src01_dst01, swu, coded (000=no, 001=yes)
src01_dst02	char(3)	src01_dst02, swu, coded (000=no, 001=yes)
src01_dst03	char(3)	src01_dst03, swu, coded (000=no, 001=yes)
src01_dst04	char(3)	src01_dst04, swu, coded (000=no, 001=yes)
src01_dst05	char(3)	src01_dst05, swu, coded (000=no, 001=yes)

src01_dst06	char(3)	src01_dst06, swu, coded (000=no, 001=yes)
src01_dst07	char(3)	src01_dst07, swu, coded (000=no, 001=yes)
src01_dst08	char(3)	src01_dst08, swu, coded (000=no, 001=yes)
src01_dst09	char(3)	src01_dst09, swu, coded (000=no, 001=yes)
src01_dst10	char(3)	src01_dst10, swu, coded (000=no, 001=yes)
src01_dst11	char(3)	src01_dst11, swu, coded (000=no, 001=yes)
src01_dst12	char(3)	src01_dst12, swu, coded (000=no, 001=yes)
src01_dst13	char(3)	src01_dst13, swu, coded (000=no, 001=yes)
src01_dst14	char(3)	src01_dst14, swu, coded (000=no, 001=yes)
src01_dst15	char(3)	src01_dst15, swu, coded (000=no, 001=yes)
src02_dst00	char(3)	src02_dst00, swu, coded (000=no, 001=yes)
src02_dst01	char(3)	src02_dst01, swu, coded (000=no, 001=yes)
src02_dst02	char(3)	src02_dst02, swu, coded (000=no, 001=yes)
src02_dst03	char(3)	src02_dst03, swu, coded (000=no, 001=yes)
src02_dst04	char(3)	src02_dst04, swu, coded (000=no, 001=yes)
src02_dst05	char(3)	src02_dst05, swu, coded (000=no, 001=yes)
src02_dst06	char(3)	src02_dst06, swu, coded (000=no, 001=yes)
src02_dst07	char(3)	src02_dst07, swu, coded (000=no, 001=yes)
src02_dst08	char(3)	src02_dst08, swu, coded (000=no, 001=yes)
src02_dst09	char(3)	src02_dst09, swu, coded (000=no, 001=yes)
src02_dst10	char(3)	src02_dst10, swu, coded (000=no, 001=yes)
src02_dst11	char(3)	src02_dst11, swu, coded (000=no, 001=yes)
src02_dst12	char(3)	src02_dst12, swu, coded (000=no, 001=yes)
src02_dst13	char(3)	src02_dst13, swu, coded (000=no, 001=yes)
src02_dst14	char(3)	src02_dst14, swu, coded (000=no, 001=yes)
src02_dst15	char(3)	src02_dst15, swu, coded (000=no, 001=yes)
src03_dst00	char(3)	src03_dst00, swu, coded (000=no, 001=yes)
src03_dst01	char(3)	src03_dst01, swu, coded (000=no, 001=yes)
src03_dst02	char(3)	src03_dst02, swu, coded (000=no, 001=yes)
src03_dst03	char(3)	src03_dst03, swu, coded (000=no, 001=yes)
src03_dst04	char(3)	src03_dst04, swu, coded (000=no, 001=yes)
src03_dst05	char(3)	src03_dst05, swu, coded (000=no, 001=yes)

src03_dst06	char(3)	src03_dst06, swu, coded (000=no, 001=yes)
src03_dst07	char(3)	src03_dst07, swu, coded (000=no, 001=yes)
src03_dst08	char(3)	src03_dst08, swu, coded (000=no, 001=yes)
src03_dst09	char(3)	src03_dst09, swu, coded (000=no, 001=yes)
src03_dst10	char(3)	src03_dst10, swu, coded (000=no, 001=yes)
src03_dst11	char(3)	src03_dst11, swu, coded (000=no, 001=yes)
src03_dst12	char(3)	src03_dst12, swu, coded (000=no, 001=yes)
src03_dst13	char(3)	src03_dst13, swu, coded (000=no, 001=yes)
src03_dst14	char(3)	src03_dst14, swu, coded (000=no, 001=yes)
src03_dst15	char(3)	src03_dst15, swu, coded (000=no, 001=yes)
src04_dst00	char(3)	src04_dst00, swu, coded (000=no, 001=yes)
src04_dst01	char(3)	src04_dst01, swu, coded (000=no, 001=yes)
src04_dst02	char(3)	src04_dst02, swu, coded (000=no, 001=yes)
src04_dst03	char(3)	src04_dst03, swu, coded (000=no, 001=yes)
src04_dst04	char(3)	src04_dst04, swu, coded (000=no, 001=yes)
src04_dst05	char(3)	src04_dst05, swu, coded (000=no, 001=yes)
src04_dst06	char(3)	src04_dst06, swu, coded (000=no, 001=yes)
src04_dst07	char(3)	src04_dst07, swu, coded (000=no, 001=yes)
src04_dst08	char(3)	src04_dst08, swu, coded (000=no, 001=yes)
src04_dst09	char(3)	src04_dst09, swu, coded (000=no, 001=yes)
src04_dst10	char(3)	src04_dst10, swu, coded (000=no, 001=yes)
src04_dst11	char(3)	src04_dst11, swu, coded (000=no, 001=yes)
src04_dst12	char(3)	src04_dst12, swu, coded (000=no, 001=yes)
src04_dst13	char(3)	src04_dst13, swu, coded (000=no, 001=yes)
src04_dst14	char(3)	src04_dst14, swu, coded (000=no, 001=yes)
src04_dst15	char(3)	src04_dst15, swu, coded (000=no, 001=yes)
src05_dst00	char(3)	src05_dst00, swu, coded (000=no, 001=yes)
src05_dst01	char(3)	src05_dst01, swu, coded (000=no, 001=yes)
src05_dst02	char(3)	src05_dst02, swu, coded (000=no, 001=yes)
src05_dst03	char(3)	src05_dst03, swu, coded (000=no, 001=yes)
src05_dst04	char(3)	src05_dst04, swu, coded (000=no, 001=yes)
src05_dst05	char(3)	src05_dst05, swu, coded (000=no, 001=yes)

src05_dst06	char(3)	src05_dst06, swu, coded (000=no, 001=yes)
src05_dst07	char(3)	src05_dst07, swu, coded (000=no, 001=yes)
src05_dst08	char(3)	src05_dst08, swu, coded (000=no, 001=yes)
src05_dst09	char(3)	src05_dst09, swu, coded (000=no, 001=yes)
src05_dst10	char(3)	src05_dst10, swu, coded (000=no, 001=yes)
src05_dst11	char(3)	src05_dst11, swu, coded (000=no, 001=yes)
src05_dst12	char(3)	src05_dst12, swu, coded (000=no, 001=yes)
src05_dst13	char(3)	src05_dst13, swu, coded (000=no, 001=yes)
src05_dst14	char(3)	src05_dst14, swu, coded (000=no, 001=yes)
src05_dst15	char(3)	src05_dst15, swu, coded (000=no, 001=yes)
src06_dst00	char(3)	src06_dst00, swu, coded (000=no, 001=yes)
src06_dst01	char(3)	src06_dst01, swu, coded (000=no, 001=yes)
src06_dst02	char(3)	src06_dst02, swu, coded (000=no, 001=yes)
src06_dst03	char(3)	src06_dst03, swu, coded (000=no, 001=yes)
src06_dst04	char(3)	src06_dst04, swu, coded (000=no, 001=yes)
src06_dst05	char(3)	src06_dst05, swu, coded (000=no, 001=yes)
src06_dst06	char(3)	src06_dst06, swu, coded (000=no, 001=yes)
src06_dst07	char(3)	src06_dst07, swu, coded (000=no, 001=yes)
src06_dst08	char(3)	src06_dst08, swu, coded (000=no, 001=yes)
src06_dst09	char(3)	src06_dst09, swu, coded (000=no, 001=yes)
src06_dst10	char(3)	src06_dst10, swu, coded (000=no, 001=yes)
src06_dst11	char(3)	src06_dst11, swu, coded (000=no, 001=yes)
src06_dst12	char(3)	src06_dst12, swu, coded (000=no, 001=yes)
src06_dst13	char(3)	src06_dst13, swu, coded (000=no, 001=yes)
src06_dst14	char(3)	src06_dst14, swu, coded (000=no, 001=yes)
src06_dst15	char(3)	src06_dst15, swu, coded (000=no, 001=yes)
src07_dst00	char(3)	src07_dst00, swu, coded (000=no, 001=yes)
src07_dst01	char(3)	src07_dst01, swu, coded (000=no, 001=yes)
src07_dst02	char(3)	src07_dst02, swu, coded (000=no, 001=yes)
src07_dst03	char(3)	src07_dst03, swu, coded (000=no, 001=yes)
src07_dst04	char(3)	src07_dst04, swu, coded (000=no, 001=yes)
src07_dst05	char(3)	src07_dst05, swu, coded (000=no, 001=yes)

src07_dst06	char(3)	src07_dst06, swu, coded (000=no, 001=yes)
src07_dst07	char(3)	src07_dst07, swu, coded (000=no, 001=yes)
src07_dst08	char(3)	src07_dst08, swu, coded (000=no, 001=yes)
src07_dst09	char(3)	src07_dst09, swu, coded (000=no, 001=yes)
src07_dst10	char(3)	src07_dst10, swu, coded (000=no, 001=yes)
src07_dst11	char(3)	src07_dst11, swu, coded (000=no, 001=yes)
src07_dst12	char(3)	src07_dst12, swu, coded (000=no, 001=yes)
src07_dst13	char(3)	src07_dst13, swu, coded (000=no, 001=yes)
src07_dst14	char(3)	src07_dst14, swu, coded (000=no, 001=yes)
src07_dst15	char(3)	src07_dst15, swu, coded (000=no, 001=yes)
src08_dst00	char(3)	src08_dst00, swu, coded (000=no, 001=yes)
src08_dst01	char(3)	src08_dst01, swu, coded (000=no, 001=yes)
src08_dst02	char(3)	src08_dst02, swu, coded (000=no, 001=yes)
src08_dst03	char(3)	src08_dst03, swu, coded (000=no, 001=yes)
src08_dst04	char(3)	src08_dst04, swu, coded (000=no, 001=yes)
src08_dst05	char(3)	src08_dst05, swu, coded (000=no, 001=yes)
src08_dst06	char(3)	src08_dst06, swu, coded (000=no, 001=yes)
src08_dst07	char(3)	src08_dst07, swu, coded (000=no, 001=yes)
src08_dst08	char(3)	src08_dst08, swu, coded (000=no, 001=yes)
src08_dst09	char(3)	src08_dst09, swu, coded (000=no, 001=yes)
src08_dst10	char(3)	src08_dst10, swu, coded (000=no, 001=yes)
src08_dst11	char(3)	src08_dst11, swu, coded (000=no, 001=yes)
src08_dst12	char(3)	src08_dst12, swu, coded (000=no, 001=yes)
src08_dst13	char(3)	src08_dst13, swu, coded (000=no, 001=yes)
src08_dst14	char(3)	src08_dst14, swu, coded (000=no, 001=yes)
src08_dst15	char(3)	src08_dst15, swu, coded (000=no, 001=yes)
src09_dst00	char(3)	src09_dst00, swu, coded (000=no, 001=yes)
src09_dst01	char(3)	src09_dst01, swu, coded (000=no, 001=yes)
src09_dst02	char(3)	src09_dst02, swu, coded (000=no, 001=yes)
src09_dst03	char(3)	src09_dst03, swu, coded (000=no, 001=yes)
src09_dst04	char(3)	src09_dst04, swu, coded (000=no, 001=yes)
src09_dst05	char(3)	src09_dst05, swu, coded (000=no, 001=yes)

src09_dst06	char(3)	src09_dst06, swu, coded (000=no, 001=yes)
src09_dst07	char(3)	src09_dst07, swu, coded (000=no, 001=yes)
src09_dst08	char(3)	src09_dst08, swu, coded (000=no, 001=yes)
src09_dst09	char(3)	src09_dst09, swu, coded (000=no, 001=yes)
src09_dst10	char(3)	src09_dst10, swu, coded (000=no, 001=yes)
src09_dst11	char(3)	src09_dst11, swu, coded (000=no, 001=yes)
src09_dst12	char(3)	src09_dst12, swu, coded (000=no, 001=yes)
src09_dst13	char(3)	src09_dst13, swu, coded (000=no, 001=yes)
src09_dst14	char(3)	src09_dst14, swu, coded (000=no, 001=yes)
src09_dst15	char(3)	src09_dst15, swu, coded (000=no, 001=yes)
src10_dst00	char(3)	src10_dst00, swu, coded (000=no, 001=yes)
src10_dst01	char(3)	src10_dst01, swu, coded (000=no, 001=yes)
src10_dst02	char(3)	src10_dst02, swu, coded (000=no, 001=yes)
src10_dst03	char(3)	src10_dst03, swu, coded (000=no, 001=yes)
src10_dst04	char(3)	src10_dst04, swu, coded (000=no, 001=yes)
src10_dst05	char(3)	src10_dst05, swu, coded (000=no, 001=yes)
src10_dst06	char(3)	src10_dst06, swu, coded (000=no, 001=yes)
src10_dst07	char(3)	src10_dst07, swu, coded (000=no, 001=yes)
src10_dst08	char(3)	src10_dst08, swu, coded (000=no, 001=yes)
src10_dst09	char(3)	src10_dst09, swu, coded (000=no, 001=yes)
src10_dst10	char(3)	src10_dst10, swu, coded (000=no, 001=yes)
src10_dst11	char(3)	src10_dst11, swu, coded (000=no, 001=yes)
src10_dst12	char(3)	src10_dst12, swu, coded (000=no, 001=yes)
src10_dst13	char(3)	src10_dst13, swu, coded (000=no, 001=yes)
src10_dst14	char(3)	src10_dst14, swu, coded (000=no, 001=yes)
src10_dst15	char(3)	src10_dst15, swu, coded (000=no, 001=yes)
src11_dst00	char(3)	src11_dst00, swu, coded (000=no, 001=yes)
src11_dst01	char(3)	src11_dst01, swu, coded (000=no, 001=yes)
src11_dst02	char(3)	src11_dst02, swu, coded (000=no, 001=yes)
src11_dst03	char(3)	src11_dst03, swu, coded (000=no, 001=yes)
src11_dst04	char(3)	src11_dst04, swu, coded (000=no, 001=yes)
src11_dst05	char(3)	src11_dst05, swu, coded (000=no, 001=yes)

src11_dst06	char(3)	src11_dst06, swu, coded (000=no, 001=yes)
src11_dst07	char(3)	src11_dst07, swu, coded (000=no, 001=yes)
src11_dst08	char(3)	src11_dst08, swu, coded (000=no, 001=yes)
src11_dst09	char(3)	src11_dst09, swu, coded (000=no, 001=yes)
src11_dst10	char(3)	src11_dst10, swu, coded (000=no, 001=yes)
src11_dst11	char(3)	src11_dst11, swu, coded (000=no, 001=yes)
src11_dst12	char(3)	src11_dst12, swu, coded (000=no, 001=yes)
src11_dst13	char(3)	src11_dst13, swu, coded (000=no, 001=yes)
src11_dst14	char(3)	src11_dst14, swu, coded (000=no, 001=yes)
src11_dst15	char(3)	src11_dst15, swu, coded (000=no, 001=yes)
src12_dst00	char(3)	src12_dst00, swu, coded (000=no, 001=yes)
src12_dst01	char(3)	src12_dst01, swu, coded (000=no, 001=yes)
src12_dst02	char(3)	src12_dst02, swu, coded (000=no, 001=yes)
src12_dst03	char(3)	src12_dst03, swu, coded (000=no, 001=yes)
src12_dst04	char(3)	src12_dst04, swu, coded (000=no, 001=yes)
src12_dst05	char(3)	src12_dst05, swu, coded (000=no, 001=yes)
src12_dst06	char(3)	src12_dst06, swu, coded (000=no, 001=yes)
src12_dst07	char(3)	src12_dst07, swu, coded (000=no, 001=yes)
src12_dst08	char(3)	src12_dst08, swu, coded (000=no, 001=yes)
src12_dst09	char(3)	src12_dst09, swu, coded (000=no, 001=yes)
src12_dst10	char(3)	src12_dst10, swu, coded (000=no, 001=yes)
src12_dst11	char(3)	src12_dst11, swu, coded (000=no, 001=yes)
src12_dst12	char(3)	src12_dst12, swu, coded (000=no, 001=yes)
src12_dst13	char(3)	src12_dst13, swu, coded (000=no, 001=yes)
src12_dst14	char(3)	src12_dst14, swu, coded (000=no, 001=yes)
src12_dst15	char(3)	src12_dst15, swu, coded (000=no, 001=yes)
src13_dst00	char(3)	src13_dst00, swu, coded (000=no, 001=yes)
src13_dst01	char(3)	src13_dst01, swu, coded (000=no, 001=yes)
src13_dst02	char(3)	src13_dst02, swu, coded (000=no, 001=yes)
src13_dst03	char(3)	src13_dst03, swu, coded (000=no, 001=yes)
src13_dst04	char(3)	src13_dst04, swu, coded (000=no, 001=yes)
src13_dst05	char(3)	src13_dst05, swu, coded (000=no, 001=yes)

src13_dst06	char(3)	src13_dst06, swu, coded (000=no, 001=yes)
src13_dst07	char(3)	src13_dst07, swu, coded (000=no, 001=yes)
src13_dst08	char(3)	src13_dst08, swu, coded (000=no, 001=yes)
src13_dst09	char(3)	src13_dst09, swu, coded (000=no, 001=yes)
src13_dst10	char(3)	src13_dst10, swu, coded (000=no, 001=yes)
src13_dst11	char(3)	src13_dst11, swu, coded (000=no, 001=yes)
src13_dst12	char(3)	src13_dst12, swu, coded (000=no, 001=yes)
src13_dst13	char(3)	src13_dst13, swu, coded (000=no, 001=yes)
src13_dst14	char(3)	src13_dst14, swu, coded (000=no, 001=yes)
src13_dst15	char(3)	src13_dst15, swu, coded (000=no, 001=yes)
src14_dst00	char(3)	src14_dst00, swu, coded (000=no, 001=yes)
src14_dst01	char(3)	src14_dst01, swu, coded (000=no, 001=yes)
src14_dst02	char(3)	src14_dst02, swu, coded (000=no, 001=yes)
src14_dst03	char(3)	src14_dst03, swu, coded (000=no, 001=yes)
src14_dst04	char(3)	src14_dst04, swu, coded (000=no, 001=yes)
src14_dst05	char(3)	src14_dst05, swu, coded (000=no, 001=yes)
src14_dst06	char(3)	src14_dst06, swu, coded (000=no, 001=yes)
src14_dst07	char(3)	src14_dst07, swu, coded (000=no, 001=yes)
src14_dst08	char(3)	src14_dst08, swu, coded (000=no, 001=yes)
src14_dst09	char(3)	src14_dst09, swu, coded (000=no, 001=yes)
src14_dst10	char(3)	src14_dst10, swu, coded (000=no, 001=yes)
src14_dst11	char(3)	src14_dst11, swu, coded (000=no, 001=yes)
src14_dst12	char(3)	src14_dst12, swu, coded (000=no, 001=yes)
src14_dst13	char(3)	src14_dst13, swu, coded (000=no, 001=yes)
src14_dst14	char(3)	src14_dst14, swu, coded (000=no, 001=yes)
src14_dst15	char(3)	src14_dst15, swu, coded (000=no, 001=yes)
src15_dst00	char(3)	src15_dst00, swu, coded (000=no, 001=yes)
src15_dst01	char(3)	src15_dst01, swu, coded (000=no, 001=yes)
src15_dst02	char(3)	src15_dst02, swu, coded (000=no, 001=yes)
src15_dst03	char(3)	src15_dst03, swu, coded (000=no, 001=yes)
src15_dst04	char(3)	src15_dst04, swu, coded (000=no, 001=yes)
src15_dst05	char(3)	src15_dst05, swu, coded (000=no, 001=yes)

src15_dst06	char(3)	src15_dst06, swu, coded (000=no, 001=yes)
src15_dst07	char(3)	src15_dst07, swu, coded (000=no, 001=yes)
src15_dst08	char(3)	src15_dst08, swu, coded (000=no, 001=yes)
src15_dst09	char(3)	src15_dst09, swu, coded (000=no, 001=yes)
src15_dst10	char(3)	src15_dst10, swu, coded (000=no, 001=yes)
src15_dst11	char(3)	src15_dst11, swu, coded (000=no, 001=yes)
src15_dst12	char(3)	src15_dst12, swu, coded (000=no, 001=yes)
src15_dst13	char(3)	src15_dst13, swu, coded (000=no, 001=yes)
src15_dst14	char(3)	src15_dst14, swu, coded (000=no, 001=yes)
src15_dst15	char(3)	src15_dst15, swu, coded (000=no, 001=yes)

10.2.1.99 VBZ_AUSWEIS

Table 122: VBZ_AUSWEIS

VBZ_AUSWEIS		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
card00_crdr00	char(3)	card00_crdr00, swu, coded (000=no,001=yes)
card00_crdr01	char(3)	card00_crdr01, swu, coded (000=no,001=yes)
card00_crdr02	char(3)	card00_crdr02, swu, coded (000=no,001=yes)
card00_crdr03	char(3)	card00_crdr03, swu, coded (000=no,001=yes)
card00_crdr04	char(3)	card00_crdr04, swu, coded (000=no,001=yes)
card00_crdr05	char(3)	card00_crdr05, swu, coded (000=no,001=yes)
card00_crdr06	char(3)	card00_crdr06, swu, coded (000=no,001=yes)
card00_crdr07	char(3)	card00_crdr07, swu, coded (000=no,001=yes)
card00_crdr08	char(3)	card00_crdr08, swu, coded (000=no,001=yes)
card00_crdr09	char(3)	card00_crdr09, swu, coded (000=no,001=yes)
card00_crdr10	char(3)	card00_crdr10, swu, coded (000=no,001=yes)
card00_crdr11	char(3)	card00_crdr11, swu, coded (000=no,001=yes)
card00_crdr12	char(3)	card00_crdr12, swu, coded (000=no,001=yes)

card00_crdr13	char(3)	card00_crdr13, swu, coded (000=no,001=yes)
card00_crdr14	char(3)	card00_crdr14, swu, coded (000=no,001=yes)
card00_crdr15	char(3)	card00_crdr15, swu, coded (000=no,001=yes)
card01_crdr00	char(3)	card01_crdr00, swu, coded (000=no,001=yes)
card01_crdr01	char(3)	card01_crdr01, swu, coded (000=no,001=yes)
card01_crdr02	char(3)	card01_crdr02, swu, coded (000=no,001=yes)
card01_crdr03	char(3)	card01_crdr03, swu, coded (000=no,001=yes)
card01_crdr04	char(3)	card01_crdr04, swu, coded (000=no,001=yes)
card01_crdr05	char(3)	card01_crdr05, swu, coded (000=no,001=yes)
card01_crdr06	char(3)	card01_crdr06, swu, coded (000=no,001=yes)
card01_crdr07	char(3)	card01_crdr07, swu, coded (000=no,001=yes)
card01_crdr08	char(3)	card01_crdr08, swu, coded (000=no,001=yes)
card01_crdr09	char(3)	card01_crdr09, swu, coded (000=no,001=yes)
card01_crdr10	char(3)	card01_crdr10, swu, coded (000=no,001=yes)
card01_crdr11	char(3)	card01_crdr11, swu, coded (000=no,001=yes)
card01_crdr12	char(3)	card01_crdr12, swu, coded (000=no,001=yes)
card01_crdr13	char(3)	card01_crdr13, swu, coded (000=no,001=yes)
card01_crdr14	char(3)	card01_crdr14, swu, coded (000=no,001=yes)
card01_crdr15	char(3)	card01_crdr15, swu, coded (000=no,001=yes)
card02_crdr00	char(3)	card02_crdr00, swu, coded (000=no,001=yes)
card02_crdr01	char(3)	card02_crdr01, swu, coded (000=no,001=yes)
card02_crdr02	char(3)	card02_crdr02, swu, coded (000=no,001=yes)
card02_crdr03	char(3)	card02_crdr03, swu, coded (000=no,001=yes)
card02_crdr04	char(3)	card02_crdr04, swu, coded (000=no,001=yes)
card02_crdr05	char(3)	card02_crdr05, swu, coded (000=no,001=yes)
card02_crdr06	char(3)	card02_crdr06, swu, coded (000=no,001=yes)
card02_crdr07	char(3)	card02_crdr07, swu, coded (000=no,001=yes)
card02_crdr08	char(3)	card02_crdr08, swu, coded (000=no,001=yes)
card02_crdr09	char(3)	card02_crdr09, swu, coded (000=no,001=yes)
card02_crdr10	char(3)	card02_crdr10, swu, coded (000=no,001=yes)
card02_crdr11	char(3)	card02_crdr11, swu, coded (000=no,001=yes)
card02_crdr12	char(3)	card02_crdr12, swu, coded (000=no,001=yes)

card02_crdr13	char(3)	card02_crdr13, swu, coded (000=no,001=yes)
card02_crdr14	char(3)	card02_crdr14, swu, coded (000=no,001=yes)
card02_crdr15	char(3)	card02_crdr15, swu, coded (000=no,001=yes)
card03_crdr00	char(3)	card03_crdr00, swu, coded (000=no,001=yes)
card03_crdr01	char(3)	card03_crdr01, swu, coded (000=no,001=yes)
card03_crdr02	char(3)	card03_crdr02, swu, coded (000=no,001=yes)
card03_crdr03	char(3)	card03_crdr03, swu, coded (000=no,001=yes)
card03_crdr04	char(3)	card03_crdr04, swu, coded (000=no,001=yes)
card03_crdr05	char(3)	card03_crdr05, swu, coded (000=no,001=yes)
card03_crdr06	char(3)	card03_crdr06, swu, coded (000=no,001=yes)
card03_crdr07	char(3)	card03_crdr07, swu, coded (000=no,001=yes)
card03_crdr08	char(3)	card03_crdr08, swu, coded (000=no,001=yes)
card03_crdr09	char(3)	card03_crdr09, swu, coded (000=no,001=yes)
card03_crdr10	char(3)	card03_crdr10, swu, coded (000=no,001=yes)
card03_crdr11	char(3)	card03_crdr11, swu, coded (000=no,001=yes)
card03_crdr12	char(3)	card03_crdr12, swu, coded (000=no,001=yes)
card03_crdr13	char(3)	card03_crdr13, swu, coded (000=no,001=yes)
card03_crdr14	char(3)	card03_crdr14, swu, coded (000=no,001=yes)
card03_crdr15	char(3)	card03_crdr15, swu, coded (000=no,001=yes)
card04_crdr00	char(3)	card04_crdr00, swu, coded (000=no,001=yes)
card04_crdr01	char(3)	card04_crdr01, swu, coded (000=no,001=yes)
card04_crdr02	char(3)	card04_crdr02, swu, coded (000=no,001=yes)
card04_crdr03	char(3)	card04_crdr03, swu, coded (000=no,001=yes)
card04_crdr04	char(3)	card04_crdr04, swu, coded (000=no,001=yes)
card04_crdr05	char(3)	card04_crdr05, swu, coded (000=no,001=yes)
card04_crdr06	char(3)	card04_crdr06, swu, coded (000=no,001=yes)
card04_crdr07	char(3)	card04_crdr07, swu, coded (000=no,001=yes)
card04_crdr08	char(3)	card04_crdr08, swu, coded (000=no,001=yes)
card04_crdr09	char(3)	card04_crdr09, swu, coded (000=no,001=yes)
card04_crdr10	char(3)	card04_crdr10, swu, coded (000=no,001=yes)
card04_crdr11	char(3)	card04_crdr11, swu, coded (000=no,001=yes)
card04_crdr12	char(3)	card04_crdr12, swu, coded (000=no,001=yes)

card04_crdr13	char(3)	card04_crdr13, swu, coded (000=no,001=yes)
card04_crdr14	char(3)	card04_crdr14, swu, coded (000=no,001=yes)
card04_crdr15	char(3)	card04_crdr15, swu, coded (000=no,001=yes)
card05_crdr00	char(3)	card05_crdr00, swu, coded (000=no,001=yes)
card05_crdr01	char(3)	card05_crdr01, swu, coded (000=no,001=yes)
card05_crdr02	char(3)	card05_crdr02, swu, coded (000=no,001=yes)
card05_crdr03	char(3)	card05_crdr03, swu, coded (000=no,001=yes)
card05_crdr04	char(3)	card05_crdr04, swu, coded (000=no,001=yes)
card05_crdr05	char(3)	card05_crdr05, swu, coded (000=no,001=yes)
card05_crdr06	char(3)	card05_crdr06, swu, coded (000=no,001=yes)
card05_crdr07	char(3)	card05_crdr07, swu, coded (000=no,001=yes)
card05_crdr08	char(3)	card05_crdr08, swu, coded (000=no,001=yes)
card05_crdr09	char(3)	card05_crdr09, swu, coded (000=no,001=yes)
card05_crdr10	char(3)	card05_crdr10, swu, coded (000=no,001=yes)
card05_crdr11	char(3)	card05_crdr11, swu, coded (000=no,001=yes)
card05_crdr12	char(3)	card05_crdr12, swu, coded (000=no,001=yes)
card05_crdr13	char(3)	card05_crdr13, swu, coded (000=no,001=yes)
card05_crdr14	char(3)	card05_crdr14, swu, coded (000=no,001=yes)
card05_crdr15	char(3)	card05_crdr15, swu, coded (000=no,001=yes)
card06_crdr00	char(3)	card06_crdr00, swu, coded (000=no,001=yes)
card06_crdr01	char(3)	card06_crdr01, swu, coded (000=no,001=yes)
card06_crdr02	char(3)	card06_crdr02, swu, coded (000=no,001=yes)
card06_crdr03	char(3)	card06_crdr03, swu, coded (000=no,001=yes)
card06_crdr04	char(3)	card06_crdr04, swu, coded (000=no,001=yes)
card06_crdr05	char(3)	card06_crdr05, swu, coded (000=no,001=yes)
card06_crdr06	char(3)	card06_crdr06, swu, coded (000=no,001=yes)
card06_crdr07	char(3)	card06_crdr07, swu, coded (000=no,001=yes)
card06_crdr08	char(3)	card06_crdr08, swu, coded (000=no,001=yes)
card06_crdr09	char(3)	card06_crdr09, swu, coded (000=no,001=yes)
card06_crdr10	char(3)	card06_crdr10, swu, coded (000=no,001=yes)
card06_crdr11	char(3)	card06_crdr11, swu, coded (000=no,001=yes)
card06_crdr12	char(3)	card06_crdr12, swu, coded (000=no,001=yes)

card06_crdr13	char(3)	card06_crdr13, swu, coded (000=no,001=yes)
card06_crdr14	char(3)	card06_crdr14, swu, coded (000=no,001=yes)
card06_crdr15	char(3)	card06_crdr15, swu, coded (000=no,001=yes)
card07_crdr00	char(3)	card07_crdr00, swu, coded (000=no,001=yes)
card07_crdr01	char(3)	card07_crdr01, swu, coded (000=no,001=yes)
card07_crdr02	char(3)	card07_crdr02, swu, coded (000=no,001=yes)
card07_crdr03	char(3)	card07_crdr03, swu, coded (000=no,001=yes)
card07_crdr04	char(3)	card07_crdr04, swu, coded (000=no,001=yes)
card07_crdr05	char(3)	card07_crdr05, swu, coded (000=no,001=yes)
card07_crdr06	char(3)	card07_crdr06, swu, coded (000=no,001=yes)
card07_crdr07	char(3)	card07_crdr07, swu, coded (000=no,001=yes)
card07_crdr08	char(3)	card07_crdr08, swu, coded (000=no,001=yes)
card07_crdr09	char(3)	card07_crdr09, swu, coded (000=no,001=yes)
card07_crdr10	char(3)	card07_crdr10, swu, coded (000=no,001=yes)
card07_crdr11	char(3)	card07_crdr11, swu, coded (000=no,001=yes)
card07_crdr12	char(3)	card07_crdr12, swu, coded (000=no,001=yes)
card07_crdr13	char(3)	card07_crdr13, swu, coded (000=no,001=yes)
card07_crdr14	char(3)	card07_crdr14, swu, coded (000=no,001=yes)
card07_crdr15	char(3)	card07_crdr15, swu, coded (000=no,001=yes)
card08_crdr00	char(3)	card08_crdr00, swu, coded (000=no,001=yes)
card08_crdr01	char(3)	card08_crdr01, swu, coded (000=no,001=yes)
card08_crdr02	char(3)	card08_crdr02, swu, coded (000=no,001=yes)
card08_crdr03	char(3)	card08_crdr03, swu, coded (000=no,001=yes)
card08_crdr04	char(3)	card08_crdr04, swu, coded (000=no,001=yes)
card08_crdr05	char(3)	card08_crdr05, swu, coded (000=no,001=yes)
card08_crdr06	char(3)	card08_crdr06, swu, coded (000=no,001=yes)
card08_crdr07	char(3)	card08_crdr07, swu, coded (000=no,001=yes)
card08_crdr08	char(3)	card08_crdr08, swu, coded (000=no,001=yes)
card08_crdr09	char(3)	card08_crdr09, swu, coded (000=no,001=yes)
card08_crdr10	char(3)	card08_crdr10, swu, coded (000=no,001=yes)
card08_crdr11	char(3)	card08_crdr11, swu, coded (000=no,001=yes)
card08_crdr12	char(3)	card08_crdr12, swu, coded (000=no,001=yes)

card08_crdr13	char(3)	card08_crdr13, swu, coded (000=no,001=yes)
card08_crdr14	char(3)	card08_crdr14, swu, coded (000=no,001=yes)
card08_crdr15	char(3)	card08_crdr15, swu, coded (000=no,001=yes)
card09_crdr00	char(3)	card09_crdr00, swu, coded (000=no,001=yes)
card09_crdr01	char(3)	card09_crdr01, swu, coded (000=no,001=yes)
card09_crdr02	char(3)	card09_crdr02, swu, coded (000=no,001=yes)
card09_crdr03	char(3)	card09_crdr03, swu, coded (000=no,001=yes)
card09_crdr04	char(3)	card09_crdr04, swu, coded (000=no,001=yes)
card09_crdr05	char(3)	card09_crdr05, swu, coded (000=no,001=yes)
card09_crdr06	char(3)	card09_crdr06, swu, coded (000=no,001=yes)
card09_crdr07	char(3)	card09_crdr07, swu, coded (000=no,001=yes)
card09_crdr08	char(3)	card09_crdr08, swu, coded (000=no,001=yes)
card09_crdr09	char(3)	card09_crdr09, swu, coded (000=no,001=yes)
card09_crdr10	char(3)	card09_crdr10, swu, coded (000=no,001=yes)
card09_crdr11	char(3)	card09_crdr11, swu, coded (000=no,001=yes)
card09_crdr12	char(3)	card09_crdr12, swu, coded (000=no,001=yes)
card09_crdr13	char(3)	card09_crdr13, swu, coded (000=no,001=yes)
card09_crdr14	char(3)	card09_crdr14, swu, coded (000=no,001=yes)
card09_crdr15	char(3)	card09_crdr15, swu, coded (000=no,001=yes)
card10_crdr00	char(3)	card10_crdr00, swu, coded (000=no,001=yes)
card10_crdr01	char(3)	card10_crdr01, swu, coded (000=no,001=yes)
card10_crdr02	char(3)	card10_crdr02, swu, coded (000=no,001=yes)
card10_crdr03	char(3)	card10_crdr03, swu, coded (000=no,001=yes)
card10_crdr04	char(3)	card10_crdr04, swu, coded (000=no,001=yes)
card10_crdr05	char(3)	card10_crdr05, swu, coded (000=no,001=yes)
card10_crdr06	char(3)	card10_crdr06, swu, coded (000=no,001=yes)
card10_crdr07	char(3)	card10_crdr07, swu, coded (000=no,001=yes)
card10_crdr08	char(3)	card10_crdr08, swu, coded (000=no,001=yes)
card10_crdr09	char(3)	card10_crdr09, swu, coded (000=no,001=yes)
card10_crdr10	char(3)	card10_crdr10, swu, coded (000=no,001=yes)
card10_crdr11	char(3)	card10_crdr11, swu, coded (000=no,001=yes)
card10_crdr12	char(3)	card10_crdr12, swu, coded (000=no,001=yes)

card10_crdr13	char(3)	card10_crdr13, swu, coded (000=no,001=yes)
card10_crdr14	char(3)	card10_crdr14, swu, coded (000=no,001=yes)
card10_crdr15	char(3)	card10_crdr15, swu,coded (000=no,001=yes)
card11_crdr00	char(3)	card11_crdr00, swu, coded (000=no,001=yes)
card11_crdr01	char(3)	card11_crdr01, swu, coded (000=no,001=yes)
card11_crdr02	char(3)	card11_crdr02, swu, coded (000=no,001=yes)
card11_crdr03	char(3)	card11_crdr03, swu, coded (000=no,001=yes)
card11_crdr04	char(3)	card11_crdr04, swu, coded (000=no,001=yes)
card11_crdr05	char(3)	card11_crdr05, swu, coded (000=no,001=yes)
card11_crdr06	char(3)	card11_crdr06, swu, coded (000=no,001=yes)
card11_crdr07	char(3)	card11_crdr07, swu, coded (000=no,001=yes)
card11_crdr08	char(3)	card11_crdr08, swu, coded (000=no,001=yes)
card11_crdr09	char(3)	card11_crdr09, swu, coded (000=no,001=yes)
card11_crdr10	char(3)	card11_crdr10, swu, coded (000=no,001=yes)
card11_crdr11	char(3)	card11_crdr11, swu, coded (000=no,001=yes)
card11_crdr12	char(3)	card11_crdr12, swu, coded (000=no,001=yes)
card11_crdr13	char(3)	card11_crdr13, swu, coded (000=no,001=yes)
card11_crdr14	char(3)	card11_crdr14, swu, coded (000=no,001=yes)
card11_crdr15	char(3)	card11_crdr15, swu, coded (000=no,001=yes)
card12_crdr00	char(3)	card12_crdr00, swu, coded (000=no,001=yes)
card12_crdr01	char(3)	card12_crdr01, swu, coded (000=no,001=yes)
card12_crdr02	char(3)	card12_crdr02, swu, coded (000=no,001=yes)
card12_crdr03	char(3)	card12_crdr03, swu, coded (000=no,001=yes)
card12_crdr04	char(3)	card12_crdr04, swu, coded (000=no,001=yes)
card12_crdr05	char(3)	card12_crdr05, swu, coded (000=no,001=yes)
card12_crdr06	char(3)	card12_crdr06, swu, coded (000=no,001=yes)
card12_crdr07	char(3)	card12_crdr07, swu, coded (000=no,001=yes)
card12_crdr08	char(3)	card12_crdr08, swu, coded (000=no,001=yes)
card12_crdr09	char(3)	card12_crdr09, swu, coded (000=no,001=yes)
card12_crdr10	char(3)	card12_crdr10, swu, coded (000=no,001=yes)
card12_crdr11	char(3)	card12_crdr11, swu, coded (000=no,001=yes)
card12_crdr12	char(3)	card12_crdr12, swu, coded (000=no,001=yes)

card12_crdr13	char(3)	card12_crdr13, swu, coded (000=no,001=yes)
card12_crdr14	char(3)	card12_crdr14, swu, coded (000=no,001=yes)
card12_crdr15	char(3)	card12_crdr15, swu, coded (000=no,001=yes)
card13_crdr00	char(3)	card13_crdr00, swu, coded (000=no,001=yes)
card13_crdr01	char(3)	card13_crdr01, swu, coded (000=no,001=yes)
card13_crdr02	char(3)	card13_crdr02, swu, coded (000=no,001=yes)
card13_crdr03	char(3)	card13_crdr03, swu, coded (000=no,001=yes)
card13_crdr04	char(3)	card13_crdr04, swu, coded (000=no,001=yes)
card13_crdr05	char(3)	card13_crdr05, swu, coded (000=no,001=yes)
card13_crdr06	char(3)	card13_crdr06, swu, coded (000=no,001=yes)
card13_crdr07	char(3)	card13_crdr07, swu, coded (000=no,001=yes)
card13_crdr08	char(3)	card13_crdr08, swu, coded (000=no,001=yes)
card13_crdr09	char(3)	card13_crdr09, swu, coded (000=no,001=yes)
card13_crdr10	char(3)	card13_crdr10, swu, coded (000=no,001=yes)
card13_crdr11	char(3)	card13_crdr11, swu, coded (000=no,001=yes)
card13_crdr12	char(3)	card13_crdr12, swu, coded (000=no,001=yes)
card13_crdr13	char(3)	card13_crdr13, swu, coded (000=no,001=yes)
card13_crdr14	char(3)	card13_crdr14, swu, coded (000=no,001=yes)
card13_crdr15	char(3)	card13_crdr15, swu, coded (000=no,001=yes)
card14_crdr00	char(3)	card14_crdr00, swu, coded (000=no,001=yes)
card14_crdr01	char(3)	card14_crdr01, swu, coded (000=no,001=yes)
card14_crdr02	char(3)	card14_crdr02, swu, coded (000=no,001=yes)
card14_crdr03	char(3)	card14_crdr03, swu, coded (000=no,001=yes)
card14_crdr04	char(3)	card14_crdr04, swu, coded (000=no,001=yes)
card14_crdr05	char(3)	card14_crdr05, swu, coded (000=no,001=yes)
card14_crdr06	char(3)	card14_crdr06, swu, coded (000=no,001=yes)
card14_crdr07	char(3)	card14_crdr07, swu, coded (000=no,001=yes)
card14_crdr08	char(3)	card14_crdr08, swu, coded (000=no,001=yes)
card14_crdr09	char(3)	card14_crdr09, swu, coded (000=no,001=yes)
card14_crdr10	char(3)	card14_crdr10, swu, coded (000=no,001=yes)
card14_crdr11	char(3)	card14_crdr11, swu, coded (000=no,001=yes)
card14_crdr12	char(3)	card14_crdr12, swu, coded (000=no,001=yes)

card14_crdr13	char(3)	card14_crdr13, swu, coded (000=no,001=yes)
card14_crdr14	char(3)	card14_crdr14, swu, coded (000=no,001=yes)
card14_crdr15	char(3)	card14_crdr15, swu, coded (000=no,001=yes)
card15_crdr00	char(3)	card15_crdr00, swu, coded (000=no,001=yes)
card15_crdr01	char(3)	card15_crdr01, swu, coded (000=no,001=yes)
card15_crdr02	char(3)	card15_crdr02, swu, coded (000=no,001=yes)
card15_crdr03	char(3)	card15_crdr03, swu, coded (000=no,001=yes)
card15_crdr04	char(3)	card15_crdr04, swu, coded (000=no,001=yes)
card15_crdr05	char(3)	card15_crdr05, swu, coded (000=no,001=yes)
card15_crdr06	char(3)	card15_crdr06, swu, coded (000=no,001=yes)
card15_crdr07	char(3)	card15_crdr07, swu, coded (000=no,001=yes)
card15_crdr08	char(3)	card15_crdr08, swu, coded (000=no,001=yes)
card15_crdr09	char(3)	card15_crdr09, swu, coded (000=no,001=yes)
card15_crdr10	char(3)	card15_crdr10, swu, coded (000=no,001=yes)
card15_crdr11	char(3)	card15_crdr11, swu, coded (000=no,001=yes)
card15_crdr12	char(3)	card15_crdr12, swu, coded (000=no,001=yes)
card15_crdr13	char(3)	card15_crdr13, swu, coded (000=no,001=yes)
card15_crdr14	char(3)	card15_crdr14, swu, coded (000=no,001=yes)
card15_crdr15	char(3)	card15_crdr15, swu, coded (000=no,001=yes)

10.2.1.100 VBZGR

Table 123: VBZGR

VBZGR		
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name	char(4)	switch_name, key
domain	char(8)	domain
info	char(300)	info
group	char(2)	group, swu, key
appl	char(2)	Application, swu
codecall	char(6)	codecall, swu

cusname	char(18)	Customer Name, swu
witness	char(12)	Witness, swu
cfw	char(3)	Supress Display on Call Forward, swu, coded
vbz_int0	char(3)	vbz_int0, swu, coded
...	char(3)	
vbz_int15	char(3)	vbz_int15, swu, coded
vbz_ank0	char(3)	vbz_ank0, swu, coded
...	char(3)	
vbz_ank15	char(3)	vbz_an15, swu, coded

10.2.1.101 VMSCOS

Table 124: VMSCOS

VMSCOS		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name, key
vmscosnum	char(2)	vmscosnum, swu, key
info	char(300)	info

10.2.1.102 VOICE_MAIL

Table 125: V0ICE_MAIL

VOICE_MAIL		
:EXPORT:SELECT		
unique_key	int	unique_key
domain	char(8)	domain
switch_name	char(4)	switch_name
extension	char(6)	extension, swu, key
vmscosnum	char(2)	vmscosnum, swu
mail_box_typ	char(1)	mail_box_typ, swu

announc_typ	char(1)	announc_typ, swu
deletion_class	char(2)	Delete class, swu
sm_autho	char(3)	sm_autho, swu, coded (see General Coded Values (79 - 96))
distr_list1	char(3)	Distribution list 1, swu, coded (000=no, 001=yes)
distr_list2	char(3)	Distribution list 2, swu, coded (000=no, 001=yes)
distr_list3	char(3)	Distribution list 3, swu, coded (000=no, 001=yes)
distr_list4	char(3)	Distribution list 4, swu, coded (000=no, 001=yes)
distr_list5	char(3)	Distribution list 5, swu, coded (000=no, 001=yes)
distr_list6	char(3)	Distribution list 6, swu, coded (000=no, 001=yes)
distr_list7	char(3)	Distribution list 7, swu, coded (000=no, 001=yes)
distr_list8	char(3)	Distribution list 8, swu, coded (000=no, 001=yes)
pin	char(12)	pin_num, swu
password	char(6)	password, swu
vms_switch	char(4)	vms_switch, swu, key

10.2.1.103 XPR_DATA

Table 126: XPR_DATA

XPR_DATA		
:EXPORT:SELECT:SELECT_UPDATES		
unique_key	int	unique_key
country	char(2)	
company	char(30)	
org1	char(40)	
org2	char(40)	
org3	char(40)	
org4	char(40)	
org5	char(40)	
org6	char(40)	
name	char(50)	
sortname	char(60)	

christianname	char(30)
title	char(15)
salutation	char(15)
location	char(15)
building	char(30)
longname	char(10)
extension	char(12)
switch_name	char(4)
sequ_num	int
tienum	char(6)
ext_areacode	char(6)
ext_netcode	char(16)
ext_ext	char(6)
faxnumber	char(22)
import_dat	char(1)
owner_group	char(30)
modify_date	char(14)
access_control	char(1)
text1	char(50)
text2	char(50)
text3	char(50)
direct_trunk_acc	char(1)
equipment	char(160)
l1	char(4)
l2	char(4)
l3	char(4)
l4	char(4)
l5	char(4)
l6	char(4)
l7	char(4)
l8	char(4)
l9	char(4)

l10	char(4)
l11	char(4)
l12	char(4)
l13	char(4)
l14	char(4)
l15	char(4)
l16	char(4)
l17	char(4)
l18	char(4)
l19	char(4)
l20	char(4)
l_prefix	char(7)
text4	char(50)
ext_countrycode	char(4)
chargeid	char(32)
basic_charge_class	char(12)
bankcode	char(15)
accountnum	char(15)
xpr_exists	char(3)
xpr_key	char(25)
xpr_pin	char(12)
xpr_server	char(25)
xpr_vmsadmin	char(3)
xpr_broadcast	char(3)
xpr_dct_send	char(3)
xpr_dct_receive	char(3)
xpr_password	char(3)
xpr_name	char(3)
xpr_flag1	char(3)
xpr_flag2	char(3)
xpr_connect	char(3)
xpr_quota	char(6)

xpr_reserve1	char(6)
xpr_reserve2	char(100)
xpr_delpwd	char(3)
persnum	char(32)
domain	char(8)
idf	char(10)
idf_cable_pair	char(10)
info_5	char(50)
info_6	char(50)
info_7	char(50)
info_8	char(50)
info_9	char(50)
info_10	char(50)
info_11	char(50)
info_12	char(50)
info_13	char(50)
info_14	char(50)
info_15	char(50)
info_16	char(50)
info_17	char(50)
info_18	char(50)
info_19	char(50)
time_update	char(22)
time_create	char(22)
displayname	char(22)

10.2.2 File Interface Objects - Detailed Description

10.2.2.1 BUILDIMP

Table 127: BUILDIMP

ODF name: Operations:	**BUILDIMP** Export: **Select** Import: **Insert, Delete, Update**

Field	Data type	Description
identifier	int	unique key
location	char(5)	location
building	char(5)	building
longname	char(25)	long name, designation

10.2.2.2 COMPIMP

Table 128: COMPIMP

ODFname: Operations:	**COMPIMP** Export: **Select** Import: **Insert, Delete, Update**	
Field	Data type	Description
identifier	int	unique key
country	char(2)	country
company	char(30)	company
address1	char(30)	address
address2	char(30)	address

10.2.2.3 COSDATA

Table 129: COSDATA

ODFname: Operations:	**COSDATA** Export: **Select** Import: **-**	
Field	Data type	Description
domain	char(8)	domain name
switch	char(4)	switch name
cosnum	char(4)	number of class of service
info	char(50)	description
c_mark_v01 - _v120	char(3)	classmarks v01 to v120
c_mark_d01 - _d51	char(3)	classmarks d01 to d51
c_mark_f01 - _f51	char(3)	classmarks f01 to f51
info2	char(50)	description part2

10.2.2.4 CTRYCODES

Table 130: CTRYCODES

ODF name: Operations:	**CTRYCODES** Export: **Select** Import: **Insert, Delete, Update**	
Field	**Data type**	**Description**
identifier	int	Unique key
src_country	char(2)	Source country
dest_country	char(2)	Destination country
id	char(2)	Destination code
info	char(50)	Description

10.2.2.5 DEVCOM

Table 131: DEVCOM

ODF name: Operations:	**DEVCOM** Export: **Select** Import: **Insert, Update, Delete**	
Field	Data type	Description
identifier	int	unique key
devconame	char(15)	name
info	char(50)	description
chargeindic	char(2)	basic charge class (CCU)
devtyp1	char(3)	device family (coded value 200, ...) (P)
devtext1	char(8)	text for devtyp1
idcard_rd	char(3)	ID card reader (coded value: yes, no)
addon_typ	char(3)	add-on device (coded value for add-on device)
addon_text	char(15)	exact device type
exact_dev_typ	char(3)	exact device type (coded value for exact device type) (P)
handsfree	char(3)	hands-free talking (coded value: yes/no)
devtyp2	char(3)	coded value for add-on device
devtext2	char(20)	text for devtyp2
devtyp3	char(3)	coded value for add-on device
devtext3	char(20)	text for devtyp3
devtyp4	char(3)	coded value for add-on device

devtext4	char(20)	text for devtyp4
devtyp5	char(3)	coded value for add-on device
devtext5	char(20)	text for devtyp5
display	char(3)	Display: 000:No, 001: Yes, 013: NUREXT
exact_dev_text	char(20)	exact device type (coded value)
adaptor_1	char(3)	adaptor 1 (coded value for adaptors - but not ISDN adaptors)
adaptor_2	char(3)	adaptor 2 (coded value for adaptors)
opticom	char(3)	ComManager (coded value: yes, no)
headset	char(3)	Headset (coded value: "No", "Without indication", "With indication")
recorder	char(3)	recorder (coded value for "Normal" or "Indication")

10.2.2.6 DOMAINDATA

Table 132: DOMAINDATA

ODF name: Operations:	**DOMAINDATA** Export: **Select** Import: **-**	
Field	Data type	Description
identifier	int	unique key
domain	char(8)	name of domain
info	char(30)	additional info text
tienum	char(6)	tie line
config_type	char(3)	numbering plan type (coded value for type1, type2)

10.2.2.7 EXTENSIONS

Table 133: EXTENSIONS

ODF name: Operations:	**EXTENSIONS** Export: **Select** Import: **-**	
Field	Data type	Description
domain	char(8)	domain name
extension	char(12)	station number
switch	char(4)	switch name
typ	char(3)	type of extension

busy	char(3)	is extension busy or free
reserved_pen	char(15)	Cabinet-Shelf-Slot-Port
reserved_switch	char(3)	reserved

10.2.2.8 GRP_PBOOK_DATA

Table 134: GRP_PBOOK_DATA

ODF name: Operations:	**GRP_PBOOK_DATA**	Export: **Select**
Field	Data type	Description
identifier_pbook	int	unique_key
identifier_brdcast	int	unique_key
identifier_sub	int	unique_key
system_name	char(16)	
grp_pbook_id	int	
dect_id	char(16)	
grp_pbook_name	char(16)	
last_changes	int	
info	char(300)	
not_send_event	char(3)	
status	char(50)	
switch_name	char(4)	
contact_name	char(16)	
contact_phone	char(32)	
contact_surname	char(16)	
contact_phone_bus	char(32)	
contact_phone_mob	char(32)	
contact_email	char(32)	
contact_birthdate	int	

10.2.2.9 HUNT

Table 135: HUNT

ODF name: Operations:	**HUNT** Export: **Select** Import: **-**	
Field	Data type	Description
domain	char(8)	switch-name
switch	char(4)	switch-name
huntgrpnum	char(4)	huntgroup number
search_criteria	char(3)	type of searching
call_sel_cap	char(3)	call selection capacity
name	char(16)	name for huntgroup at the display
mailbox	char(6)	extension number of mailbox
grouptype	char(3)	master huntgroup ? (yes/no)
callforward	char(3)	call-forwarding ? (yes/no)
add_ext	char(6)	group-index
vario	char(3)	variable call selector (yes/no)
extension	char(6)	extension
position	int	position within a hunt group
info	char(50)	description
overflow_type	char(3)	overflow type
overflow_ext	char(22)	overflow extension
overflow_loop	char(3)	overflow after one cycle
customer	char(3)	customer specific announcement sync
cfd_net	char(3)	call forwarding net
remote_switch	char(4)	Remote system
remote_domain	char(8)	Remote domain

10.2.2.10 LOCIMP

Table 136: LOCIMP

ODF name: Operations:	**LOCIMP** Export: **Select** Import: **Insert, Delete, Update**	
Field	Data type	Description
identifier	int	unique key

location	char(5)	location
name	char(20)	name
address1	char(30)	address
address2	char(30)	address

10.2.2.11 ORGIMP

Table 137: ORGIMP

ODF name: Operations:	**ORGIMP** Export: **Select** Import: **Insert, Delete, Update**	
Field	Data type	Description
identifier	int	unique key
country	char(2)	country (P)
company	char(30)	company (P)
org1	char(12)	department (P, if org2 & org3 are blank)
org2	char(12)	department (P, if org2 & org3 are blank)
org3	char(12)	department (P, if org2 & org3 are blank)

10.2.2.12 PENDATA

Table 138: PENDATA

ODF name: Operations:	**PENDATA** Export: **Select**	
Field	Data type	Description
identifier	int	unique key
domain	char(8)	domain
switch	char(4)	switch name
pen	char(11)	physical equipment number (LTG/LTU/EBT/SATZ)
cardtyp	char(3)	type of board
status	char(3)	is this pen administered with a device (free/busy)
info	char(50)	description
node_1	char(6)	main distribution frame_1 (Knoten)
subnode_1	char(2)	main distribution frame_1 (Unterknoten)
line_1	char(3)	main distribution frame_1 (Bucht)

strip_1	char(2)	main distribution frame_1 (Leiste)
pin_1	char(3)	main distribution frame_1 (Stift)
node_2	char(6)	main distribution frame_2 (Knoten)
subnode_2	char(2)	main distribution frame_2 (Unterknoten)
line_2	char(3)	main distribution frame_2 (Bucht)
strip_2	char(2)	main distribution frame_2 (Leiste)
pin_2	char(3)	main distribution frame_2 (Stift)
ctr_for_busy	int	number of assigned S0-bus subscriber
reserved	char(3)	is this pen free or reserved
extension	char(6)	extension to this pen

10.2.2.13 PERS

Table 139: PERS

ODF name: Operations:	**PERS** Export: **Select, Select_updates** Import: **Insert, Delete, Update**	
Field	Data type	Description
identifier	int	unique key
country	char(2)	country (short code) (P)
company	char(30)	company
org1	char(40)	department
org2	char(40)	department
org3	char(40)	department
name	char(50)	name of the subscriber (P)
sortname	char(60)	name without muted vowels/special characters
christianname	char(30)	first name
title	char(15)	title
salutation	char(15)	address
location	char(15)	location
building	char(30)	building
longname	char(10)	room
extension	char(12)	internal extension
switch	char(4)	Hicom switch

position	int	position
tienum	char(6)	tie line
ext_areacode	char(6)	area code
ext_netcode	char(16)	CO number
ext_ext	char(6)	extension
faxnumber	char(22)	fax number
owner_group	char(30)	ETD user group
direct_trunk_acc	char(1)	flag (for displaying trunk access)
text1	char(24)	key word 1
text2	char(24)	key word 2
text3	char(24)	key word 3
text4	char(24)	key word 4
import_dat	char(1)	import data
access_control	char(1)	action control
modify_date	char(14)	modification date
equipment	char(160)	device equipment
l1	char(4)	service type 1
l2	char(4)	service type 2
l3	char(4)	service type 3
l4	char(4)	service type 4
l5	char(4)	service type 5
l6	char(4)	service type 6
l7	char(4)	service type 7
l8	char(4)	service type 8
l9	char(4)	service type 9
l10	char(4)	service type 10
l11	char(4)	service type 11
l12	char(4)	service type 12
l13	char(4)	service type 13
l14	char(4)	service type 14
l15	char(4)	service type 15
l16	char(4)	service type 16

l17	char(4)	service type 17
l18	char(4)	service type 18
l19	char(4)	service type 19
l20	char(4)	service type 20
l_prefix	char(7)	service prefix
ext_countrycode	char(4)	ext. country code
costbear	char(15)	cost centre
chargeid	char(6)	basic charge class
bankcode	char(15)	bank sort code
accountnum	char(15)	account number

10.2.2.14 PICKUP

Table 140: PICKUP

ODF name: Operations:	**PICKUP** Export: **Select, Select_updates** Import: **-**	
Field	Data type	Description
domain	char(8)	domain name
switch	char(4)	switch name
pickupgrpnum	char(4)	pickup group number
attcall	char(3)	attention call, coded
displ	char(3)	shown on display
secondcall	char(3)	pickup if second call too
extension	char(6)	extension
info	char(50)	description of pickup group
remote_subscriber	char(3)	remote subscriber

10.2.2.15 PINDATA

Table 141: PINDATA

ODF name: Operations:	**PINDATA** Export: **Select, Select_updates** Import: **Insert, Update, Delete**	
Field	Data type	Description
identifier	int	unique key

domain	char(8)	domain
extension	char(12)	extension
switch	char(4)	switch
pin_num	char(12)	PIN
pin_type	char(3)	type (coded value: MANUAL, IDCARD, BUCODE)
position	int	position (link to PERS via extension, switch)
pin_class	char(2)	PIN class
pinindiv	char(3)	

10.2.2.16 PORTDATA

Table 142: PORTDATA

ODF name: Operations:	**PORTDATA** Export: **Select, Select_updates**Import: Insert, Delete,Update	
Field	Data type	Description
identifier	int	unique key
extension	char(6)	extension
switch	char(4)	switch
pen	char(13)	physical equipment number
devconame	char(20)	name of the assigned device combination
opstate	char(3)	operation state (000=no, 001=yes, 008=card, 056=logged off)
chargeid	char(6)	billing account
wabegrpnum	char(2)	dial plan group number
vbzgrpnum	char(2)	internal traffic restriction group
tcos	char(4)	class of service 1
ncos	char(4)	class of service 2
berumgrpnum	char(2)	class of service changeover group
stbnum	char(2)	standard key
bkrnl	char(3)	basic speed dial list
speed_dial	char(3)	individual speed dial list (coded value for yes/no)
speed_dial_two	char(2)	speed dial list 2
numdisp	char(3)	display numbers (coded value for yes, no)

devtyp	char(3)	voice terminal device type
devtext	char(15)	text for devtyp
att_tone_source	char(3)	attention tone for source subscriber
att_tone_target	char(3)	attention tone for target subscriber
call_back_free	char(3)	callback - no answer override
call_back_busy	char(3)	callback - busy override
displayname	char(31)	not provided, use field "name" in ODF DISPLAY_NAME for export/import or ODF PORT_DISP for export !
dl_idx_man	char(3)	download class (manual down-loading)
dl_idx_auto	char(3)	download class (automatic down-loading)
domain	char(8)	domain
least_cost_rout1	char(4)	least cost routing 1 (0 = no lcr)
least_cost_rout2	char(4)	least cost routing 2 (0 = no lcr)
least_cost_data1	char(4)	lcr data 1
least_cost_data2	char(4)	lcr data 2
navar	char(3)	night service option (coded value for yes, no)
assoc	char(3)	associated subscriber (coded value for yes, no)
analog_modem	char(3)	analog side modem (coded value for yes, no)
clc_num	char(3)	CLC (coded value for yes, no)
bus_type	char(3)	bus type (coded value for mbus, mbus etc.)
bus_extension	char(6)	main extension (S0 bus)
tsi_voice	char(1)	TSI Voice
tsi_data	char(1)	TSI Data
tsi_fax	char(1)	TSI FAX
protocol_voice	char(3)	Protocol voice (coded value for 1TR6,...)
protocol_data	char(3)	Protocol data (coded value for 1TR6,...)
protocol_fax	char(3)	Protocol FAX (coded value for 1TR6,...)
msn	char(3)	MSN (coded value for yes, no)
tsi_ttx	char(1)	TSI TTX
tsi_btx	char(1)	TSI BTX
tsi_msn	char(1)	TSI MSN
connect_type	char(3)	connection type (coded value for connection type)

opt_voice	char(2)	option index voice (AMO ZAND)
opt_data	char(2)	option index data
opt_fax	char(2)	option index fax
info	char(50)	description
secr_device	char(3)	subscriber is a secretary (yes/no)
fcfw	char(22)	extension for fixed call forwarding
registration	char(3)	subscriber registered for Cordless E (coded value: yes, no)
opt_btx	char(2)	option for VTX (1..10)
opt_ttx	char(2)	option for TTX (1..10)
prot_btx	char(3)	protocol for VTX (coded value for SB1TR61, SB1TR62, SBQ9311, SBQ9312, SBDKZ1, SBDKZ2, SBEDSR1, SBEDSR2, SBEDSR3, SBEDSR4)
prot_ttx	char(3)	protocol for TTX (as for prot_btx)
p_akl	char(3)	Camp-on authorisation (coded value: yes, no)
p_aufaks	char(3)	override/camp-on rejection (coded value: yes, no)
p_dirans	char(3)	voice calling (coded value: yes, no)
p_hot	char(3)	hotline (coded value: yes, no)
p_natr	char(3)	authorisation for emergency override/release (coded value: yes, no)
p_roe	char(3)	code-blue/off-hook recall (coded value: yes, no)
p_agent	char(3)	ACD agent (coded value: yes, no)
p_autolog	char(3)	automatic logon of ACD agent (coded value: yes, no)
p_caf	char(3)	ACD agent feature control (coded value: yes, no)
p_castmon	char(3)	can activate monitoring (silent/tone monitoring) (coded value: yes, no)
p_cbstmon	char(3)	monitoring possible (silent/tone monitoring) (coded value: yes, no)
p_dndsr	char(3)	suppress alerting tone for agent/super (coded value: yes, no)
p_super	char(3)	ACD supervisor (coded value: yes, no)
p_ani1	char(3)	automatic number identification category 1 (coded value: yes, no)
p_ani2	char(3)	automatic number identification category 2 (coded value: yes, no)

p_ani3	char(3)	automatic number identification category 3 (coded value: yes, no)
p_ani4	char(3)	automatic number identification category 4 (coded value: yes, no)
p_ani5	char(3)	automatic number identification category 5 (coded value: yes, no)
p_ani6	char(3)	automatic number identification category 6 (coded value: yes, no)
p_ani7	char(3)	automatic number identification category 7 (coded value: yes, no)
p_ani8	char(3)	automatic number identification category 8 (coded value: yes, no)
p_ani9	char(3)	automatic number identification category 9 (coded value: yes, no)
p_ani10	char(3)	automatic number identification category 10 (coded value: yes, no)
net_config	char(3)	network-wide configuration (coded value: yes, no)
log_pty_switch	char(4)	logical station switch
log_pty_ext	char(6)	logical station extension
log_pty_pen	char(13)	logical station PEN
rem_pty_switch	char(4)	remote station switch
rem_pty_ext	char(6)	remote station extension
rem_pty_pen	char(13)	remote station PEN
extend_pty_ext	char(6)	extension connection extension
ext_pty_ptnr_ext	char(6)	partner data extension
ext_pty_ptnr_ext_l	char(22)	partner data secondary extension
ext_pty_ptnr_swi	char(4)	partner data switch
opt_ntvlv	char(2)	option net team extension connection (values from 1 to 10)
prot_ntvlv	char(3)	protocol net team extension connection (coded value for protocol types)
permakt	char(3)	Permanently activated
cordless	char(3)	Cordless
lwpar	char(2)	Loadware parameter
uart	char(3)	Call forwarding type
num_callback	char(2)	Max. number of callbacks

protocol_spp	char(3)	S0PP/S2PP protocol
opt_spp	char(2)	S0PP/S2PP option
alarm	char(4)	Alarm number
circidx	char(3)	CIRCIDX
cofidx	char(2)	COFIDX
call_free_mail	char(3)	Callback on free
language	char(3)	Preferred language
p_sprstat	char(3)	Voice static
p_sprdyn	char(3)	Voice dynamic
vpi	char(1)	VPI
vci	char(5)	VCI
status	char(26)	Status
call_log	char(3)	call journal: A06:none, C18:all, 011:attempts
api	char(3)	API: 000:no, 001:yes
p_spawh	char(3)	SPAWH: 000:no, 001:yes
p_spsc	char(3)	SPSC: 000:no, 001:yes
p_spenb	char(3)	SPEND: 000:no, 001:yes
auto_privacy	char(3)	automatic private call: 000:no, 001:yes
single_button_mode	char(3)	single button mode: 000:no, 001:yes
out_line_sel	char(3)	preference: H32:free, A06:none, H31:last, H30:primary
inc_line_sel	char(3)	preference: H33:calling, H34:calling (primary), A06:none, H35:incoming, H36:incoming (primary)
ring_type	char(3)	alerting tone when station busy: H38: alerting tone, H37:no ringing
key_system	char(3)	key system
busindex	int	busindex

10.2.2.17 PORT_DISP (port/display_name Join)

Table 143: PORT_DISP

ODF name: Operations:	**PORT_DISP (PORT/DISPLAY_NAME Join)** Export: **Select** Import: **-**	
Field	Data type	Description
unique_key	int	unique_key

extension	char(12)	extension, swu, key
switch_name	char(4)	switch_name, key
pen	char(13)	pen, swu
devconame	char(20)	Device combination
opstate	char(3)	operation state, swu, coded (000=no, 001=yes, 008=card, 056=logged off)
chargeid	char(6)	chargeid
wabegrpnum	char(2)	wabegrpnum, swu
vbzgrpnum	char(2)	vbzgrpnum, swu
tcos	char(4)	Cos 1, swu
ncos	char(4)	Cos 2, swu
berumgrpnum	char(2)	berumgrpnum, swu
stbnum	char(3)	num_key_lay, swu
bkrnl	char(2)	Speed dial list 1, swu
speed_dial_two	char(2)	Speed dial list 2, swu
speed_dial	char(3)	speed_dial, swu, coded
numdisp	char(3)	displ, swu, coded
devtyp	char(3)	device type, swu, coded
devtext	char(20)	devtext, swu
att_tone_source	char(3)	VC warn tone source, swu, coded
att_tone_target	char(3)	VC warn tone destination, swu, coded
technical_ext	char(6)	Technical extension, swu
switch_new	char(4)	switch_new, swu
temporary	char(3)	Temporary extension, coded
call_back_free	char(3)	Callback-no answer rejection, swu, coded
call_back_busy	char(3)	Callback-busy rejection, swu, coded
displayname	char(31)	Display name, swu, from display_name table
department	char(25)	Department, swu, from display_name table
position	int	Index, swu, key, from display_name table
charcon	char(3)	charcon, swu, coded, from display_name table
dl_idx_man	char(3)	Manual downloading, swu
dl_idx_auto	char(3)	Automatic downloading, swu

domain	char(8)	domain
least_cost_rout1	char(2)	LCRCOS 1, swu
least_cost_rout2	char(2)	LCRCOS 2, swu
least_cost_data1	char(2)	LCRCOSDAT 1, swu
least_cost_data2	char(2)	LCRCOSDAT 2, swu
navar	char(3)	navar, swu, coded
assoc	char(3)	Associated subscriber, swu, coded
analog_modem	char(3)	analogue_modem, swu, coded
clc_num	char(3)	CLC number, swu, coded
bus_type	char(3)	bus_type, swu, coded
bus_extension	char(6)	bus_extension, swu
tsi_voice	char(1)	TSI voice, swu
tsi_data	char(1)	TSI data, swu
tsi_fax	char(1)	TSI fax, swu
protocol_voice	char(3)	Protocol voice, swu, coded
protocol_data	char(3)	Protocol data, swu, coded
protocol_fax	char(3)	Protocol fax, swu, coded
msn	char(3)	msn, swu, coded
tsi_ttx	char(1)	TSI ttx, swu
tsi_btx	char(1)	TSI vtx, swu
tsi_msn	char(1)	TSI msn, swu
connect_type	char(3)	connect_type, swu, coded
opt_voice	char(2)	Voice option, swu
opt_data	char(2)	Data option, swu
opt_fax	char(2)	Fax option, swu
secr_device	char(3)	Secretary telephone, swu, coded
info	char(50)	info
fcfw	char(22)	fcfw, swu
compl_mod_allowed	char(3)	compl_mod_allowed, swu, coded
registration	char(3)	Registration, swu, coded
opt_btx	char(2)	VTX option, swu
opt_ttx	char(2)	TTX option, swu

prot_btx	char(3)	VTX protocol, swu, coded
prot_ttx	char(3)	TTX protocol, swu, coded
p_akl	char(3)	KN, swu, coded
p_aufaks	char(3)	RKNOVR, swu, coded
p_aufs	char(3)	KNOVR, swu, coded
p_dirans	char(3)	VC, swu, coded
p_hot	char(3)	HOT, swu, coded
p_natr	char(3)	EOVR, swu, coded
p_roe	char(3)	OFHKRC, swu, coded
p_agent	char(3)	AGENT, swu, coded
p_autolog	char(3)	AUTOLOG, swu, coded
p_caf	char(3)	CAF, swu, coded
p_castmon	char(3)	CASTMON, swu, coded
p_cbstmon	char(3)	CBSTMON, swu, coded
p_dndsr	char(3)	DNDSR, swu, coded
p_super	char(3)	SUPER, swu, coded
p_ani1	char(3)	ANI1, swu, coded
p_ani2	char(3)	ANI2, swu, coded
p_ani3	char(3)	ANI3, swu, coded
p_ani4	char(3)	ANI4, swu, coded
p_ani5	char(3)	ANI5, swu, coded
p_ani6	char(3)	ANI6, swu, coded
p_ani7	char(3)	ANI7, swu, coded
p_ani8	char(3)	ANI8, swu, coded
p_ani9	char(3)	ANI9, swu, coded
p_ani10	char(3)	ANI10, swu, coded
pen_num	int	pen_num, swu
net_config	char(3)	Network-Wide Configuration, coded
log_pty_switch	char(4)	Logical Subscriber System, swu
log_pty_ext	char(6)	Logical Subscriber Number, swu
log_pty_pen	char(13)	Logical Subscriber PEN, swu
log_pty_pen_num	int	log_pty_pen_num, swu

rem_pty_switch	char(4)	Remote Subscriber System, swu
rem_pty_ext	char(6)	Remote Subscriber Number, swu
rem_pty_pen	char(13)	Remote Subscriber PEN, swu
rem_pty_pen_num	int	rem_pty_pen_num, swu
extend_pty_ext	char(6)	Extended Connection Number, swu
ext_pty_ext_l	char(22)	Extension number of extended call, swu
ext_pty_ptnr_ext	char(6)	Destination Data Number
ext_pty_ptnr_ext_l	char(22)	Destination Data Extension Number, swu
ext_pty_ptnr_swi	char(4)	Destination Data System
conf_not_consist	char(3)	conf_not_consist, swu, coded
opt_ntvlv	char(2)	NTVLV Option, swu
prot_ntvlv	char(3)	NTVLV Protocol, swu, coded
permakt	char(3)	permakt, swu, coded
cordless	char(3)	cordless, swu, coded
lwpar	char(2)	lwpar, swu
uart	char(3)	uart, swu, coded
num_callback	char(2)	Max. Number of Callbacks, swu
protocol_spp	char(3)	Protocol S0PP/S2PP, swu, coded
opt_spp	char(2)	Option S0PP/S2PP, swu
alarm	char(4)	Alarm Number, swu
circidx	char(3)	CIRCIDX, swu
cofidx	char(2)	COFIDX, swu
call_free_mail	char(3)	Callback on No Answer, swu, coded
language	char(3)	Preferred Language, swu, coded
p_sprstat	char(3)	Language Static, swu, coded
p_sprdyn	char(3)	Language Dynamic, swu, coded
vpi	char(1)	VPI, swu
vci	char(5)	VCI, swu
status	char(26)	Status, swu
call_log	char(3)	Call Logging, swu, coded (A06:none, C18:all, 011:attempts)
api	char(3)	Work Station Protocol, swu, coded

p_spawh	char(3)	LOADF, swu, coded
p_spsc	char(3)	LOSC, swu, coded
p_spenb	char(3)	LOENB, swu, coded
auto_privacy	char(3)	Automatic Privacy, swu, coded
single_button_mode	char(3)	Single Key Mode, swu, coded
out_line_sel	char(3)	Outgoing Call Preference, swu, coded
inc_line_sel	char(3)	Incoming Call Preference, swu, coded
ring_type	char(3)	Subscriber busy warning tone, swu, coded
key_system	char(3)	Key System, swu, coded
busindex	int	busindex, swu
p_teamchef	char(3)	Executive, swu, coded
p_teamsecr	char(3)	Secretary, swu, coded
p_drziel	char(3)	DSS Dest. networkwide, swu, coded
teamnum	char(3)	Exec./Secr. Team, swu
teamfunct	char(3)	Team Function, swu, coded
store_calls	char(3)	Store calls, swu, coded
dtb_access	char(3)	DTB Access, swu, coded
dual_mode_skey	char(3)	Dual Mode Softkey, swu, coded
sms_feature	char(3)	SMS, swu, coded
reg_counter	char(1)	Registration counter, swu
ic_data_type	char(3)	Data Type, swu, coded
ltu	char(2)	ltu
trace_mal	char(3)	malicious call trace allowed, swu, coded
asynct	char(4)	asynchronious time, swu
offtype	char(3)	switching off type, swu, coded
hmusic	char(3)	music on hold recording, swu
extbus	char(3)	extented bus, swu, coded
pattern	char(3)	idlecode pattern for the b-channels, swu
pulslev	char(3)	level of charge metering pulses, swu, coded
pulstype	char(3)	type of charge metering pulses, swu, coded
ackst	char(3)	acknowledge state between dps and announcement device, swu, coded

exact_dev_typ	char(3)	exact device type, swu, coded
exact_dev_text	char(20)	remarks to exact device type, swu
ccmscd	char(6)	multiple code-calling code, swu
ccno	char(6)	code-calling number, swu
spec_outpulse_dev	char(3)	spec_outpulse_dev, swu, coded
special	char(3)	special station, swu, coded
sufdial	char(3)	station can suffix dial, swu, coded
witness	char(3)	witness, swu, coded
phone_mail_idx	char(2)	index for PhoneMail / Xpression access number, swu
dnidsp	char(3)	display DNIS text, swu, coded
circuit	char(3)	circuit number
flash_enable	char(3)	enable hook flash, swu, coded
ops_conn_type	char(3)	Off Premise connection type, swu, coded
src_grp_idx	char(2)	source group index, swu
classmrk_ec	char(3)	classmrk_ec, swu, coded
classmrk_g711	char(3)	classmrk_g711, swu, coded
classmrk_g729	char(3)	classmrk_g729, swu, coded
classmrk_vad	char(3)	classmrk_vad, swu, coded
classmrk_init	char(3)	classmrk_init, swu, coded
hotline_idx	char(3)	Hotline Index, swu
node_id	char(10)	node_id, swu
cdracc	char(12)	CDR Account Code, swu
fwd_spk_call	char(3)	fwd_spk_call, swu, coded
p_spk1way	char(3)	p_spk1way, swu, coded
com_group	char(4)	com_group, swu
com_spk_call_2wy	char(3)	com_spk_call_2wy, swu, coded
spk_call_1wy_brd	char(3)	spk_call_1wy_brd, swu, coded
p_elapsed_time	char(3)	Elapsed Time, swu, coded
p_block_auto_rel	char(3)	p_block_auto_rel, swu, coded
protocol_class	char(3)	protocol_class, swu, coded
lin1	char(22)	lin1, swu
lin2	char(22)	lin2, swu

acf_appl	char(3)	acf_appl, swu
dtmf_conf	char(3)	dtmf_conf, swu, coded
dhpar_asend	char(3)	dhpar_asend, swu, coded
dhpar_bsend	char(3)	dhpar_bsend, swu, coded
dhpar_dtmfst	char(3)	dhpar_dtmfst, swu, coded
dhpar_flash	char(3)	dhpar_flash, swu, coded
dhpar_vmsdp	char(3)	dhpar_vmsdp, swu, coded
dhpar_ledsign	char(3)	dhpar_ledsign, swu, coded
dhpar_mwled	char(3)	dhpar_mwled, swu, coded
dhpar_predia	char(3)	dhpar_predia, swu, coded
dhpar_ring1	char(3)	dhpar_ring1, swu, coded
bas_svc_audio31k	char(3)	bas_svc_audio31k, swu, coded
bas_svc_voice	char(3)	bas_svc_voice, swu, coded
bas_svc_udi64	char(3)	bas_svc_udi64, swu, coded
bas_svc_udi64ra	char(3)	bas_svc_udi64ra, swu, coded
sztype_atnd	char(3)	sztype_atnd, swu, coded
sztype_dszco	char(3)	sztype_dszco, swu, coded
sztype_dszint	char(3)	sztype_dszint, swu, coded
sztype_expr	char(3)	sztype_expr, swu, coded
sztype_fwd	char(3)	sztype_fwd, swu, coded
sztype_fwdnans	char(3)	sztype_fwdnans, swu, coded
szcd_atnd	char(4)	szcd_atnd, swu
szcd_dszco	char(4)	szcd_dszco, swu
szcd_dszint	char(4)	szcd_dszint, swu
szcd_expr	char(4)	szcd_expr, swu
szcd_fwd	char(4)	szcd_fwd, swu
szcd_fwdnans	char(4)	szcd_fwdnans, swu
pubnum	char(15)	pubnum, swu
ton	char(3)	ton, swu, coded
npi	char(3)	npi, swu, coded
dtmf_control	char(3)	dtmf_control, swu, coded
delay_cfw_busy	char(3)	delay_cfw_busy, swu, coded

p_cfnr	char(3)	p_cfnr, swu, coded
p_dsfwdext	char(3)	p_dsfwdext, swu, coded
p_mlpp_routine	char(3)	p_mlpp_routine, swu, coded
p_mlpp_prty	char(3)	p_mlpp_prty, swu, coded
p_mlpp_immed	char(3)	p_mlpp_immed, swu, coded
p_mlpp_flash	char(3)	p_mlpp_flash, swu, coded
p_mlpp_flashov	char(3)	p_mlpp_flashov, swu, coded
p_mlpp_prot	char(3)	p_mlpp_prot, swu, coded
p_mlpp_mlpps	char(3)	p_mlpp_mlpps, swu, coded

10.2.2.18 PORTUPD

Table 144: PORTUPD

ODF name: Operations:	**PORTUPD** Export: **Select** Import: **Update**	
Field	Data type	Description
extension	char(12)	extension
switch_name	char(4)	Hicom switch
name	char(31)	display name (from DISPLAY_NAME)

10.2.2.19 SWITCHDATA

Table 145: SWITCHDATA

ODF name: Operations:	**SWITCHDATA** Export: **Select** Import: **-**	
Field	Data type	Description
identifier	int	unique key
switch	char(4)	Hicom switch
typ	char(10)	system-internal Hicom ID
version	char(5)	search criterion
info	char(50)	info text
country	char(2)	abbreviated country code
areacode	char(16)	area code of the PABX
netcode	char(16)	CO number of the PABX

amo_lang	char(3)	AMO language used (coded value for German and English)
nodenum	char(10)	unique node number in the Hicom V3.3 network
destnum	char(3)	destination number
domain	char(8)	domain name
vms_server	char(3)	position within a hunt group
vms_switch	char(4)	description
admin	char(3)	overflow type (coded value for overflow type)
lcr_data	char(3)	overflow extension
lcr_admin	char(3)	overflow after loop (coded value: yes, no)
sec_node	char(10)	customer-specific announcement (coded value: yes, no)
language1	char(3)	language1
language2	char(3)	language2
language3	char(3)	language3
language4	char(3)	language4
language5	char(3)	language5
countrycode	char(4)	countrycode
uplo2_aps	char(13)	call forward-all calls (coded value: yes, no)
uplol_aps	char(13)	remote switch
uplvm_aps	char(13)	remote domain
knotenkz	char(22)	knotenkz
consistent	char(3)	consistent

10.2.3 Assistant Objects - Detailed Description

10.2.3.1 ACD_AGENT

Table 146: ACD_AGENT

ACD_AGENT		
:EXPORT:SELECT		
unique_key	int	unique key/eindeutiger Schlüssel
switch_name	char(4)	Switchname/Anlage, key

domain	char(8)	domain/Domäne
logon_id	char(6)	logon_id, swu, key
info	char(300)	info/Beschreibung
acd_grp_asgn	char(3)	swu, coded
acd_grp_num	char(3)	swu
acd_grp_type	char(3)	swu, coded
autolg_extension	char(12)	swu
autowk_mode	char(3)	swu, coded
autowk_timr	char(3)	swu
pen	char(15)	swu
pen_num	int	pen_num,swu
bchannel	char(2)	swu
position	char(4)	swu

10.2.3.2 ACD_ANNOUNC

Table 147: ACD_ANNOUNC

ACD_ANNOUNC		
:EXPORT:SELECT		
unique_key	int	unique key/eindeutiger Schlüssel
switch_name	char(4)	Switch/Anlage, key
domain	char(8)	domain name/ Domäne
acd_anc_num	char(3)	swu
info	char(300)	info/Beschreibung
anc_length	char(2)	swu
op_mode	char(3)	swu, coded
pen	char(15)	swu, key
pen_num	int	pen_num

10.2.3.3 ACD_GROUP

Table 148: ACD_GROUP

ACD_GROUP		
:EXPORT:SELECT		
unique_key	int	unique key/eindeutiger Schlüssel
switch_name	char(4)	Switch/Anlage, key
domain	char(8)	domain/Domäne
acd_grp_num	char(3)	swu, key
info	char(300)	Info/Beschreibung
search_method	char(3)	swu, coded
call_wtg_enbld	char(3)	swu, coded
pm_svc_access_num	char(22)	swu
led_flash	char(3)	swu
led_on	char(3)	swu
led_wink	char(3)	swu
acd_grp_type	char(3)	swu, coded
sup_extension	char(12)	swu
sup_primary_grp	char(3)	swu, coded

10.2.3.4 ACD_MUSIC

Table 149: ACD_MUSIC

ACD_MUSIC		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	Switch / Anlage, key
domain	char(8)	domain / Domäne
acd_mus_num	char(2)	swu
info	char(300)	info / Beschreibung
pen	char(15)	swu, key
pen_num	int	pen number

10.2.3.5 ACD_NUMBER

Table 150: ACD_NUMBER

ACD_NUMBER		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
acd_num	char(12)	swu, key
info	char(300)	info / Beschreibung
collect_calls	char(3)	swu, coded
aud_source_id	char(3)	swu
threshold_option	char(3)	swu, coded
max_num_calls	char(4)	swu
queue_pri_ovfl	char(2)	swu
phone_display	char(24)	swu
queue_pri_prim	char(2)	swu
rtg_option	char(3)	swu, coded
rcg_num	char(4)	swu
rtg_rule_num	char(3)	swu

10.2.3.6 ACD_RCG

Table 151: ACD_RCG

ACD_RCG		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
rcg_num	char(4)	swu, key
info	char(300)	info / Beschreibung
fwd_num	char(6)	swu
shift_set_mon	char(4)	swu

shift_set_tue	char(4)	swu
shift_set_wed	char(4)	swu
shift_set_thu	char(4)	swu
shift_set_fri	char(4)	swu
shift_set_sat	char(4)	swu
shift_set_sun	char(4)	swu

10.2.3.7 ACD_RCG_SUB

Table 152: ACD_RCG_SUB

ACD_RCG_SUB		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
rcg_num	char(4)	swu, key
info	char(300)	info / Beschreibung
start_day	char(2)	swu, key
start_month	char(2)	swu, key
start_year	char(4)	swu, key
end_day	char(2)	swu
end_month	char(2)	swu
end_year	char(4)	swu
recurrent	char(3)	swu, coded
shift_set_num	char(4)	swu

10.2.3.8 ACD_RTG_TABLE

Table 153: ACD_RTG_TABLE

ACD_RTG_TABLE		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel

switch_name	char(4)	switch /Anlage, key
domain	char(8)	domain / Domäne
rtg_table_num	char(4)	swu, key
info	char(300)	info / Beschreibung
max_step	int	swu

10.2.3.9 ACD_RTG_TABLE_SUB

Table 154: ACD_RTG_TABLE_SUB

ACD_RTG_TABLE_SUB		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
rtg_table_num	char(4)	swu, key
step_num	char(2)	swu, key
action	char(3)	swu, coded
cgoto_step_num	char(2)	swu
parameter	char(22)	swu
test	char(3)	swu, coded
value	char(6)	swu
max_step	int	swu

10.2.3.10 ACD_SHIFT_SET

Table 155: ACD_SHIFT_SET

ACD_SHIFT_SET		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
shift_set_num	char(4)	swu, key

info	char(300)	info / Beschreibung

10.2.3.11 ACD_SHIFT_SET_SUB

Table 156: ACD_SHIFT_SET_SUB

ACD_SHIFT_SET_SUB		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
shift_set_num	char(4)	swu, key
start_time	char(5)	swu
end_time	char(5)	swu, key
rtg_table_num	char(4)	swu
end_of_shift_rtg	char(3)	swu, coded
info	char(300)	info

10.2.3.12 ACD_SYS_PARAM

Table 157: ACD_SYS_PARAM

ACD_SYS_PARAM		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
dial_plan	char(2)	swu
agt_display_id	char(3)	swu, coded
agt_id_length	char(1)	swu
agt_target	char(3)	swu, coded
delay_rngbck_timr	char(2)	swu
msg_emergency	char(24)	swu
call_pri_upg_timer	char(3)	swu

call_pri_upg_enbld	char(3)	swu, coded
silent_mon_enabled	char(3)	swu, coded
tone_mon_enabled	char(3)	swu, coded
serv_out_feat	char(3)	swu, coded
mon_con_dly_timer	char(3)	swu, coded
mon_pin	char(3)	swu, coded
msg_agt_sup1	char(24)	swu
msg_agt_sup2	char(24)	swu
msg_agt_sup3	char(24)	swu
msg_agt_sup4	char(24)	swu
msg_agt_sup5	char(24)	swu
msg_agt_sup6	char(24)	swu
msg_agt_sup7	char(24)	swu
msg_agt_sup8	char(24)	swu
msg_sup_agt1	char(24)	swu
msg_sup_agt2	char(24)	swu
msg_sup_agt3	char(24)	swu
msg_sup_agt4	char(24)	swu
msg_sup_agt5	char(24)	swu
msg_sup_agt6	char(24)	swu
msg_sup_agt7	char(24)	swu
msg_sup_agt8	char(24)	swu
nholdt	char(3)	nholdt

10.2.3.13 BOARD

Table 158: BOARD

BOARD		
:EXPORT:SELECT		
unique_key	int	unique_key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne

ltg	char(1)	ltg, swu, key
ltu	char(2)	ltu, swu
slot	char(3)	Slot, swu, key
pen_num_board	int	pen_num_board, swu
info	char(300)	Description / Beschreibung
part_num	char(12)	Part number / Sachnummer, swu
board_name	char(10)	Board Name / BG-Name, swu, coded
func_id	char(2)	Function ID / Funktions-ID, swu
category	char(8)	category / Kategorie, swu
loadw_var	char(1)	loadw_var, swu
num_circ	char(3)	num_circ, swu
thr_short	char(4)	thr_short, swu
thr_long	char(4)	thr_long, swu
thr_perm	char(4)	thr_perm, swu
hwy_bundle	char(3)	hwy_bundle, swu,coded
act_flow	char(3)	act_flow, swu,coded
enable_globaldata	char(3)	enable_globaldata, swu, coded
ip_address	char(15)	ip_address, swu
tc_port_no	char(5)	tc_port_no, swu
waml_port_no	char(5)	waml_port_no, swu
mac_check	char(3)	mac_check, swu, coded
ext_lan	char(3)	ext_lan, swu, coded
hw_interface	char(3)	hw_interface, swu, coded
trace_level	char(3)	trace_level, swu
isdn_station_no	char(22)	isdn_station_no, swu
hi_watermark	char(3)	hi_watermark, swu
hi_time	char(3)	hi_time, swu
low_watermark	char(2)	low_watermark, swu
low_time	char(3)	low_time, swu
dial_attempts	char(3)	dial_attempts, swu
transm_rate	char(4)	transm_rate, swu
num_packets	char(5)	num_packets, swu

pause	char(3)	pause, swu
isdn_adp_access	char(3)	isdn_adp_access, swu
lan_adp_access	char(3)	lan_adp_access, swu
isdn1_check	char(3)	isdn1_check, swu
srcip_check	char(3)	srcip_check, swu
isdn_check	char(3)	isdn_check, swu
act_atlantic	char(3)	act_atlantic, swu
act_isdn	char(3)	act_isdn, swu
enable_atmdata	char(3)	enable_atmdata, swu
interface_mode	char(3)	interface_mode, swu
protocol	char(3)	protocol, swu
reassembly_buff	char(3)	reassembly_buff, swu
vci_off0	char(3)	vci_off0, swu
vci_off1	char(3)	vci_off1, swu
vci_off2	char(3)	vci_off2, swu
vci_off3	char(3)	vci_off3, swu
vci_off4	char(3)	vci_off4, swu
vci_off5	char(3)	vci_off5, swu
vci_off6	char(3)	vci_off6, swu
vci_off7	char(3)	vci_off7, swu
numbering_plan	char(3)	numbering_plan, swu,coded
atm_addr	char(15)	atm_addr, swu
idle_bit_pattern	char(3)	idle_bit_pattern, swu
signal_vpi	char(1)	signal_vpi, swu,
vpci0	char(5)	vpci0, swu
vpci1	char(5)	vpci1, swu
vpci2	char(5)	vpci2, swu
vpci3	char(5)	vpci3, swu
vpci4	char(5)	vpci4, swu
vpci5	char(5)	vpci5, swu
vpci6	char(5)	vpci6, swu
vpci7	char(5)	vpci7, swu

loadw_param1	char(1)	loadw_param1, swu
loadw_param2	char(1)	loadw_param2, swu
file_index1	char(2)	file_index1, swu
file_index2	char(2)	file_index2, swu
op_mode	char(3)	op_mode, swu,coded
siu_mode1	char(3)	siu_mode1, swu
siu_mode2	char(3)	siu_mode2, swu
siu_mode3	char(3)	siu_mode3, swu
siu_mode4	char(3)	siu_mode4, swu
siu_mode5	char(3)	siu_mode5, swu
siu_mode6	char(3)	siu_mode6, swu
siu_mode7	char(3)	siu_mode7, swu
siu_mode8	char(3)	siu_mode8, swu
t1_loadw	char(3)	t1_loadw, swu
assign_station	char(3)	assign_station, swu
perhw_loadw_var	char(1)	perhw_loadw_var, swu
conn_type	char(3)	conn_type, swu
location	char(48)	location, swu
locid	char(3)	locid, swu
frame_type	char(3)	frame_type, swu
ltu_overlay	char(2)	ltu_overlay, swu
cbm_conf1	char(3)	cbm_conf1, swu
cbm_conf2	char(3)	cbm_conf2, swu
cbm_conf3	char(3)	cbm_conf3, swu
cbm_conf4	char(3)	cbm_conf4, swu
jitter_buff_mode	char(3)	jitter_buff_mode, swu, coded
avg_delay_vce	char(3)	swu
max_delay_vce	char(3)	swu
min_delay_vce	char(2)	swu
pack_loss_pref	char(2)	swu
avg_delay_data	char(3)	swu
max_delay_data	char(3)	swu

phone_num	char(22)	phone_num, swu
fax_num	char(22)	fax_num, swu
b_channel_count	char(3)	b_channel_count, swu
nbcs_idle_pattern	char(3)	nbcs_idle_pattern, swu
gateway_ip_addr	char(15)	gateway_ip_addr, swu
dsp_g711_pkt_size	char(2)	dsp_g711_pkt_size, swu
dsp_g729_pkt_size	char(2)	dsp_g729_pkt_size, swu
jitter_buf_depth	char(3)	jitter_buf_depth, swu
ncui_file_acc	char(3)	ncui_file_acc, swu
cmi_usr_message	char(3)	cmi_usr_message, swu
debugdet_bit0	char(3)	debugdet_bit0, swu
debugdet_bit1	char(3)	debugdet_bit1, swu
debugdet_bit2	char(3)	debugdet_bit2, swu
debugdet_bit3	char(3)	debugdet_bit3, swu
debugdet_bit4	char(3)	debugdet_bit4, swu
debugdet_bit5	char(3)	debugdet_bit5, swu
debugdet_bit6	char(3)	debugdet_bit6, swu
debugdet_bit7	char(3)	debugdet_bit7, swu
debugdet_bit8	char(3)	debugdet_bit8, swu
debugdet_bit9	char(3)	debugdet_bit9, swu
debugdet_bit10	char(3)	debugdet_bit10, swu
debugdet_bit11	char(3)	debugdet_bit11, swu
debugdet_bit12	char(3)	debugdet_bit12, swu
debugdet_bit13	char(3)	debugdet_bit13, swu
debugdet_bit14	char(3)	debugdet_bit14, swu
debugdet_bit15	char(3)	debugdet_bit15, swu
debugdet_bit16	char(3)	debugdet_bit16, swu
tffs_bit0	char(3)	tffs_bit0, swu
hhs_msg_file_size	char(4)	hhs_msg_file_size, swu
ncui_err_file_size	char(4)	ncui_err_file_size, swu
switch_file_size	char(4)	switch_file_size, swu
trace_file_size	char(4)	trace_file_size, swu

event_file_size	char(4)	event_file_size, swu
enable_autonegot	char(3)	enable_autonegot, swu
speed	char(3)	speed, swu
mode	char(3)	mode, swu
vlan_id	char(4)	vlan_id, swu
source_ip_addr	char(15)	source_ip_addr, swu
ip_net_mask	char(15)	ip_net_mask, swu
virtual_lan_tag	char(3)	virtual_lan_tag, swu
default_router	char(15)	default_router, swu
tcp_port	char(5)	tcp_port, swu
tos_payload	char(3)	tos_payload, swu
udp_range_lo	char(5)	udp_range_lo, swu
udp_range_hi	char(5)	udp_range_hi, swu
tcp_range_lo	char(5)	tcp_range_lo, swu
tcp_range_hi	char(5)	tcp_range_hi, swu
voice_detect	char(3)	voice_detect, swu
jitter_buf_adapt	char(3)	jitter_buf_adapt, swu
echo_cancel	char(3)	echo_cancel, swu
q931_response	char(5)	q931_response, swu
q931_connect	char(5)	q931_connect, swu
gateway_name	char(16)	gateway_name, swu
total_kbyte	char(5)	total_kbyte, swu
enable_g723_comp	char(3)	enable_g723_comp, swu
wait_time	char(2)	wait_time, swu
download_asc	char(3)	download_asc, swu
download_dsp	char(3)	download_dsp, swu
download_debug	char(3)	download_debug, swu
download_h323	char(3)	download_h323, swu
bw_within_lan	char(5)	bw_within_lan, swu
snmp_cs1	char(32)	snmp_cs1
snmp_cs2	char(32)	snmp_cs2
standby_mode	char(3)	standby_mode, swu

max_b_channel	char(3)	max_b_channel, swu
dsp_g723_pkt_size	char(2)	dsp_g723_pkt_size, swu
tos_signaling	char(3)	tos_signaling, swu
codec_g711	char(3)	codec_g711, swu
codec_g723	char(3)	codec_g723, swu
codec_g729	char(3)	codec_g729, swu
action	char(3)	action, swu
ref_clock_off	char(3)	ref_clock_off, swu
dmc_conn_num	char(3)	swu
ltu2	char(2)	swu
slot2	char(3)	swu
restart_part_no	char(10)	swu
restart_func_id	char(3)	swu
customer_lan_ip	char(15)	swu
customer_lan_port	char(5)	swu
atlantic_lan_ip	char(15)	swu
atlantic_lan_port	char(5)	swu
stmi2igw_sub_mask	char(15)	swu
gateway_no	char(4)	swu
gateway_ip	char(15)	swu
gateway_dir_no	char(22)	swu
pri_gatekp_ip	char(15)	swu
pri_gatekp_port	char(5)	swu
pri_gatekp_id1	char(64)	swu
pri_gatekp_id2	char(64)	swu
secnd_gatekp_ip	char(15)	swu
secnd_gatekp_port	char(5)	swu
secnd_gatekp_id1	char(64)	swu
secnd_gatekp_id2	char(64)	swu
time_to_live	char(5)	swu
man_station_ip	char(15)	swu
man_station_port	char(5)	swu

backup_server_ip	char(15)	swu
backup_server_port	char(5)	swu
gateway_ip_port	char(5)	swu
wbm_passwd	char(20)	swu
wbm_role	char(3)	swu
global_gateway_id1	char(64)	swu
global_gateway_id2	char(64)	swu
time_window	char(3)	swu
global_passwd1	char(79)	swu
global_passwd2	char(64)	swu
global_passwd3	char(64)	swu
global_passwd4	char(64)	swu
gateway_id1	char(64)	swu
gateway_id2	char(64)	swu
gateway_sect_no	char(3)	swu
default_gw_ip	char(15)	swu
ftp_login	char(60)	swu
ftp_password	char(60)	swu
tmd_file_index	char(2)	swu
h235_security	char(3)	swu
time_wndw	char(3)	swu
primary_codec	char(4)	swu
secondary_codec	char(4)	swu
tertiary_codec	char(4)	swu
download_jb	char(3)	swu, coded
download_snmp	char(3)	swu, coded
download_h235	char(3)	swu, coded
trk_prot	char(3)	
gw_aut_req	char(3)	
gw_sec_ret	char(20)	
gw_user_id	char(20)	
gw_realm	char(20)	

sip_reg	char(3)	
reg_ip1	char(15)	
reg_port1	int	
reg_time	int	
secure	char(3)	
voip_sec	char(3)	
reg_ip2	char(15)	
reg_port2	int	
global_id_hfa	char(24)	
h235_sec_igw	char(3)	
fct_blk	char(2)	
bchl_3530	char(3)	
bchl_sip	char(3)	
bchl_3550	char(3)	
bchl_waml	char(3)	
bchl_3570	char(3)	
cha_res_ln	char(3)	
trunk_prot_sip	char(3)	
trunk_prot_sipq	char(3)	
trunk_prot_h323	char(3)	
trunk_prot_h323a	char(3)	
passw1	char(19)	
passw2	char(19)	
login_term	char(10)	
reg_ext_gk	char(3)	
alarm_no	char(3)	
tls_port	char(5)	swu
dns_ip_address	char(15)	swu
dls_ip_address	char(15)	swu
dls_port	char(5)	swu
dls_accept_pass	char(3)	swu, coded
security_hfa	char(3)	swu, coded

security_h323	char(3)	swu, coded
port_tls1	char(5)	swu
port_tls2	char(5)	swu
port_tcp1	char(5)	swu
port_tcp2	char(5)	swu
license_agree	char(3)	
boardname_request	char(10)	
partnum_request	char(12)	
hightraffic_syst	char(3)	
priority_suggest	char(2)	
lunaall_suggest	char(2)	
lunamin_suggest	char(2)	
slot_suggest	char(3)	
ltu_min	char(2)	
ltu_max	char(2)	
ltu_suggested	char(2)	
usr_password	char(19)	
t38_fax	char(3)	
fmo_rfc2833	char(3)	
dtmf_rfc2833	char(3)	
red_trns_rfc2833	char(3)	
codec_prio1	char(6)	
codec_prio2	char(6)	
codec_prio3	char(6)	
codec_prio4	char(6)	
codec_prio5	char(6)	
codec_prio6	char(6)	
codec_prio7	char(6)	
vad_prio1	char(3)	
vad_prio2	char(3)	
vad_prio3	char(3)	
vad_prio4	char(3)	

vad_prio5	char(3)
vad_prio6	char(3)
vad_prio7	char(3)
rtp_size_prio1	char(2)
rtp_size_prio5	char(2)
rtp_size_prio3	char(2)
rtp_size_prio4	char(2)
rtp_size_prio5	char(2)
rtp_size_prio6	char(2)
rtp_size_prio7	char(2)
ldmcprxy	char(3)
sg_active	char(3)
siuc	char(3)
siuann	char(3)
dtr	char(3)
ipmode	char(3)
dhcpipV4	char(3)
dhcpipV6	char(3)
stdby_ip	char(15)
stdby_net_mask	char(15)
dns_ip_addres2	char(15)
mac_address	char(17)
tdmlaw	char(3)
ui_mode	char(3)
man_lan_ip_adr	char(15)
man_lan_netmask	char(15)
man_lan_vlan_tag	char(3)
man_lan_vlan_id	char(4)
man_lan_def_gw_ip	char(15)

10.2.3.14 UBGDAT

Table 159: UBGDAT

UBGDAT		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain name
part_number	char(12)	swu, key
func_id	char(2)	swu, key
board_name	char(10)	swu
thr_short	char(4)	swu
thr_long	char(4)	swu
thr_perm	char(4)	swu
circ_max	char(4)	swu
category	char(8)	category
board_type	int	swu

10.2.3.15 COP

Table 160: COP

COP		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
copnum	char(3)	copnum, swu, key
info	char(300)	info
par_aant	char(3)	swu, coded
par_aban	char(3)	swu, coded
par_abpd	char(3)	swu, coded
par_ackb	char(3)	swu, coded
par_ackf	char(3)	swu, coded

par_alul	char(3)	swu, coded
par_anic	char(3)	swu, coded
par_anio	char(3)	swu, coded
par_aniw	char(3)	swu, coded
par_annc	char(3)	swu, coded
par_ans	char(3)	swu, coded
par_appr	char(3)	swu, coded
par_aseg	char(3)	swu, coded
par_asnm	char(3)	swu, coded
par_ast1	char(3)	swu, coded
par_ast2	char(3)	swu, coded
par_br64	char(3)	swu, coded
par_btn1	char(3)	swu, coded
par_btn2	char(3)	swu, coded
par_btn3	char(3)	swu, coded
par_btof	char(3)	swu, coded
par_ccm	char(3)	swu, coded
par_ccr	char(3)	swu, coded
par_cctn	char(3)	swu, coded
par_cocn	char(3)	swu, coded
par_coei	char(3)	swu, coded
par_cott	char(3)	swu, coded
par_ctgf	char(3)	swu, coded
par_ditw	char(3)	swu, coded
par_dt1a	char(3)	swu, coded
par_dt2a	char(3)	swu, coded
par_dtbr	char(3)	swu, coded
par_dtm1	char(3)	swu, coded
par_dtm2	char(3)	swu, coded
par_dtm3	char(3)	swu, coded
par_dtm4	char(3)	swu, coded
par_dtmf	char(3)	swu, coded

par_dtn	char(3)	swu, coded
par_dtt	char(3)	swu, coded
par_dtv	char(3)	swu, coded
par_emov	char(3)	swu, coded
par_eo10	char(3)	swu, coded
par_eod	char(3)	swu, coded
par_eoda	char(3)	swu, coded
par_fdgt	char(3)	swu, coded
par_icac	char(3)	swu, coded
par_idp1	char(3)	swu, coded
par_idp2	char(3)	swu, coded
par_idp3	char(3)	swu, coded
par_idp4	char(3)	swu, coded
par_ilrw	char(3)	swu, coded
par_imex	char(3)	swu, coded
par_ivsn	char(3)	swu, coded
par_lai8	char(3)	swu, coded
par_lao0	char(3)	swu, coded
par_la03	char(3)	swu, coded
par_lknq	char(3)	swu, coded
par_lsi0	char(3)	swu, coded
par_lsi3	char(3)	swu, coded
par_lso0	char(3)	swu, coded
par_lso3	char(3)	swu, coded
par_lsrb	char(3)	swu, coded
par_lsup	char(3)	swu, coded
par_lwcc	char(3)	swu, coded
par_l3ar	char(3)	swu, coded
par_naat	char(3)	swu, coded
par_nbar	char(3)	swu, coded
par_no1a	char(3)	swu, coded
par_noan	char(3)	swu, coded

par_noch	char(3)	swu, coded
par_nona	char(3)	swu, coded
par_noto	char(3)	swu, coded
par_notr	char(3)	swu, coded
par_nsdl	char(3)	swu, coded
par_nton	char(3)	swu, coded
par_pdp1	char(3)	swu, coded
par_pdp2	char(3)	swu, coded
par_pdp3	char(3)	swu, coded
par_pdp4	char(3)	swu, coded
par_pnio	char(3)	swu, coded
par_rblk	char(3)	swu, coded
par_rcla	char(3)	swu, coded
par_rf0c	char(3)	swu, coded
par_rf1c	char(3)	swu, coded
par_rf2c	char(3)	swu, coded
par_rf3c	char(3)	swu, coded
par_rgsz	char(3)	swu, coded
par_rlsa	char(3)	swu, coded
par_rlsb	char(3)	swu, coded
par_rmsg	char(3)	swu, coded
par_rrst	char(3)	swu, coded
par_r162	char(3)	swu, coded
par_r163	char(3)	swu, coded
par_r164	char(3)	swu, coded
par_r165	char(3)	swu, coded
par_r166	char(3)	swu, coded
par_r167	char(3)	swu, coded
par_r168	char(3)	swu, coded
par_r169	char(3)	swu, coded
par_r170	char(3)	swu, coded
par_r171	char(3)	swu, coded

par_r172	char(3)	swu, coded
par_r173	char(3)	swu, coded
par_r174	char(3)	swu, coded
par_r175	char(3)	swu, coded
par_r176	char(3)	swu, coded
par_r177	char(3)	swu, coded
par_r178	char(3)	swu, coded
par_r179	char(3)	swu, coded
par_r180	char(3)	swu, coded
par_r181	char(3)	swu, coded
par_r182	char(3)	swu, coded
par_r183	char(3)	swu, coded
par_r184	char(3)	swu, coded
par_r185	char(3)	swu, coded
par_r186	char(3)	swu, coded
par_r187	char(3)	swu, coded
par_r188	char(3)	swu, coded
par_r189	char(3)	swu, coded
par_r190	char(3)	swu, coded
par_r191	char(3)	swu, coded
par_sack	char(3)	swu, coded
par_sdl	char(3)	swu, coded
par_sedl	char(3)	swu, coded
par_sfrm	char(3)	swu, coded
par_spd	char(3)	swu, coded
par_sptr	char(3)	swu, coded
par_tded	char(3)	swu, coded
par_tie	char(3)	swu, coded
par_tim1	char(3)	swu, coded
par_tim2	char(3)	swu, coded
par_tim3	char(3)	swu, coded
par_tim4	char(3)	swu, coded

par_trac	char(3)	swu, coded
par_wot2	char(3)	swu, coded
service_info	char(50)	swu
source	char(3)	swu, coded
trunk_rta	char(3)	swu, coded
trunk_rta1	char(3)	swu, coded
trunk_rta2	char(3)	swu, coded
trunk_rta3	char(3)	swu, coded
trunk_rta4	char(3)	swu, coded
trunk_rta5	char(3)	swu, coded
trunk_rta6	char(3)	swu, coded
trunk_ta	char(3)	swu, coded
toll_rta	char(3)	swu, coded
toll_rta1	char(3)	swu, coded
toll_rta2	char(3)	swu, coded
toll_rta3	char(3)	swu, coded
toll_rta4	char(3)	swu, coded
toll_rta5	char(3)	swu, coded
toll_rta6	char(3)	swu, coded
toll_ta	char(3)	swu, coded
trunk_dev_type	char(3)	swu, coded
cop_overwrite	char(3)	swu, coded
country_var	char(3)	swu, codedu
par_lcos	char(3)	swu, codedu
par_rdig	char(3)	swu, coded
par_pdig	char(3)	swu, coded
par_prmtsgn	char(3)	swu, coded
par_mchi	char(3)	swu, coded
par_uu7b	char(3)	swu, coded
par_zmfv	char(3)	swu, coded

10.2.3.16 COT

Table 161: COT

COT		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
cotnum	char(3)	swu, key
info	char(300)	info
par_aacd	char(3)	swu, coded
par_abna	char(3)	swu, coded
par_abpd	char(3)	swu, coded
par_acrc	char(3)	swu, coded
par_actr	char(3)	swu, coded
par_adiu	char(3)	swu, coded
par_aeod	char(3)	swu, coded
par_ai1	char(3)	swu, coded
par_ai1n	char(3)	swu, coded
par_ai2	char(3)	swu, coded
par_ai2n	char(3)	swu, coded
par_ai3	char(3)	swu, coded
par_ai3n	char(3)	swu, coded
par_aicf	char(3)	swu, coded
par_amfc	char(3)	swu, coded
par_anht	char(3)	swu, coded
par_anic	char(3)	swu, coded
par_anna	char(3)	swu, coded
par_annc	char(3)	swu, coded
par_anod	char(3)	swu, coded
par_ans	char(3)	swu, coded
par_aocc	char(3)	swu, coded

par_appr	char(3)	swu, coded
par_apsm	char(3)	swu, coded
par_astn	char(3)	swu, coded
par_atdi	char(3)	swu, coded
par_atrs	char(3)	swu, coded
par_bccc	char(3)	swu, coded
par_bcne	char(3)	swu, coded
par_bloc	char(3)	swu, coded
par_brar	char(3)	swu, coded
par_brla	char(3)	swu, coded
par_bsht	char(3)	swu, coded
par_catr	char(3)	swu, coded
par_cbbn	char(3)	swu, coded
par_cbfn	char(3)	swu, coded
par_ccro	char(3)	swu, coded
par_cctn	char(3)	swu, coded
par_cdbo	char(3)	swu, coded
par_cdn	char(3)	swu, coded
par_cdrd	char(3)	swu, coded
par_cebc	char(3)	swu, coded
par_ceco	char(3)	swu, coded
par_ceoc	char(3)	swu, coded
par_cfbn	char(3)	swu, coded
par_cfos	char(3)	swu, coded
par_cftr	char(3)	swu, coded
par_cfva	char(3)	swu, coded
par_chrt	char(3)	swu, coded
par_coex	char(3)	swu, coded
par_cotn	char(3)	swu, coded
par_cps6	char(3)	swu, coded
par_csn7	char(3)	swu, coded
par_ctal	char(3)	swu, coded

par_ctln	char(3)	swu, coded
par_ctls	char(3)	swu, coded
par_cug	char(3)	swu, coded
par_cwbs	char(3)	swu, coded
par_dcpa	char(3)	swu, coded
par_dfnn	char(3)	swu, coded
par_disn	char(3)	swu, coded
par_dit	char(3)	swu, coded
par_diud	char(3)	swu, coded
par_di1	char(3)	swu, coded
par_di1n	char(3)	swu, coded
par_di2	char(3)	swu, coded
par_di2n	char(3)	swu, coded
par_di3	char(3)	swu, coded
par_di3n	char(3)	swu, coded
par_dpre	char(3)	swu, coded
par_dsdl	char(3)	swu, coded
par_dsit	char(3)	swu, coded
par_dtmp	char(3)	swu, coded
par_dtms	char(3)	swu, coded
par_dtne	char(3)	swu, coded
par_dtni	char(3)	swu, coded
par_edba	char(3)	swu, coded
par_eodl	char(3)	swu, coded
par_ete	char(3)	swu, coded
par_ete1	char(3)	swu, coded
par_ete2	char(3)	swu, coded
par_ete3	char(3)	swu, coded
par_ete4	char(3)	swu, coded
par_ete5	char(3)	swu, coded
par_etsc	char(3)	swu, coded
par_fax1	char(3)	swu, coded

par_fax2	char(3)	swu, coded
par_fnan	char(3)	swu, coded
par_fwdn	char(3)	swu, coded
par_fwdr	char(3)	swu, coded
par_ibsy	char(3)	swu, coded
par_iczl	char(3)	swu, coded
par_iczo	char(3)	swu, coded
par_idis	char(3)	swu, coded
par_idnd	char(3)	swu, coded
par_ievt	char(3)	swu, coded
par_ifr	char(3)	swu, coded
par_igid	char(3)	swu, coded
par_iicb	char(3)	swu, coded
par_iidl	char(3)	swu, coded
par_inau	char(3)	swu, coded
par_indg	char(3)	swu, coded
par_ions	char(3)	swu, coded
par_irto	char(3)	swu, coded
par_itb	char(3)	swu, coded
par_itrc	char(3)	swu, coded
par_itrn	char(3)	swu, coded
par_itt	char(3)	swu, coded
par_ivac	char(3)	swu, coded
par_kndi	char(3)	swu, coded
par_knex	char(3)	swu, coded
par_knor	char(3)	swu, coded
par_lbpr	char(3)	swu, coded
par_linc	char(3)	swu, coded
par_lino	char(3)	swu, coded
par_lprn	char(3)	swu, coded
par_lrpm	char(3)	swu, coded
par_lwnc	char(3)	swu, coded

par_maov	char(3)	swu, coded
par_mfc	char(3)	swu, coded
par_mfob	char(3)	swu, coded
par_micc	char(3)	swu, coded
par_miih	char(3)	swu, coded
par_mlpr	char(3)	swu, coded
par_mrca	char(3)	swu, coded
par_msir	char(3)	swu, coded
par_naat	char(3)	swu, coded
par_ncdr	char(3)	swu, coded
par_nchd	char(3)	swu, coded
par_nchg	char(3)	swu, coded
par_nct	char(3)	swu, coded
par_nctp	char(3)	swu, coded
par_nin2	char(3)	swu, coded
par_nito	char(3)	swu, coded
par_nlcr	char(3)	swu, coded
par_nlht	char(3)	swu, coded
par_nlrd	char(3)	swu, coded
par_nmlt	char(3)	swu, coded
par_noft	char(3)	swu, coded
par_noho	char(3)	swu, coded
par_norr	char(3)	swu, coded
par_nosd	char(3)	swu, coded
par_noto	char(3)	swu, coded
par_novc	char(3)	swu, coded
par_npis	char(3)	swu, coded
par_nppr	char(3)	swu, coded
par_npun	char(3)	swu, coded
par_nqnp	char(3)	swu, coded
par_nqpi	char(3)	swu, coded
par_nraa	char(3)	swu, coded

par_nrba	char(3)	swu, coded
par_ntbo	char(3)	swu, coded
par_nton	char(3)	swu, coded
par_ohkq	char(3)	swu, coded
par_ovra	char(3)	swu, coded
par_pdob	char(3)	swu, coded
par_pdot	char(3)	swu, coded
par_pinr	char(3)	swu, coded
par_pnev	char(3)	swu, coded
par_pri	char(3)	swu, coded
par_prov	char(3)	swu, coded
par_rcbs	char(3)	swu, coded
par_rcl	char(3)	swu, coded
par_rcls	char(3)	swu, coded
par_rddi	char(3)	swu, coded
par_rlin	char(3)	swu, coded
par_ropt	char(3)	swu, coded
par_r181	char(3)	swu, coded
par_r232	char(3)	swu, coded
par_r233	char(3)	swu, coded
par_r249	char(3)	swu, coded
par_r250	char(3)	swu, coded
par_r251	char(3)	swu, coded
par_r252	char(3)	swu, coded
par_r253	char(3)	swu, coded
par_r254	char(3)	swu, coded
par_r255	char(3)	swu, coded
par_sanh	char(3)	swu, coded
par_satc	char(3)	swu, coded
par_sdid	char(3)	swu, coded
par_seco	char(3)	swu, coded
par_siri	char(3)	swu, coded

par_snbe	char(3)	swu, coded
par_spah	char(3)	swu, coded
par_spcm	char(3)	swu, coded
par_spsr	char(3)	swu, coded
par_supn	char(3)	swu, coded
par_t4wa	char(3)	swu, coded
par_tcc	char(3)	swu, coded
par_tcet	char(3)	swu, coded
par_tgre	char(3)	swu, coded
par_tie	char(3)	swu, coded
par_tiei	char(3)	swu, coded
par_tmfc	char(3)	swu, coded
par_tndv	char(3)	swu, coded
par_trai	char(3)	swu, coded
par_trba	char(3)	swu, coded
par_trid	char(3)	swu, coded
par_trsc	char(3)	swu, coded
par_tr6t	char(3)	swu, coded
par_tscs	char(3)	swu, coded
par_twds	char(3)	swu, coded
par_us1e	char(3)	swu, coded
par_usbt	char(3)	swu, coded
par_uscc	char(3)	swu, coded
par_uscd	char(3)	swu, coded
par_usd	char(3)	swu, coded
par_uus1	char(3)	swu, coded
par_uus2	char(3)	swu, coded
par_uus3	char(3)	swu, coded
par_vcmn	char(3)	swu, coded
par_vm	char(3)	swu, coded
par_vnco	char(3)	swu, coded
par_waan	char(3)	swu, coded

par_xfer	char(3)	swu, coded
par_xtrc	char(3)	swu, coded
service_info	char(50)	swu
source	char(3)	swu, coded
trunk_dev_type	char(3)	swu, coded
cot_overwrite	char(3)	swu, coded
country_var	char(3)	swu
par_bfsr	char(3)	swu, coded
par_bkna	char(3)	swu, coded
par_dtmf	char(3)	swu, coded
par_hgtr	char(3)	swu, coded
par_ibba	char(3)	swu, coded
par_orrn	char(3)	swu, coded
par_pgep	char(3)	swu, coded
par_satr	char(3)	swu, coded
par_spcp	char(3)	swu, coded
par_r258	char(3)	swu, coded
par_r259	char(3)	swu, coded
par_r260	char(3)	swu, coded
par_r261	char(3)	swu, coded
par_r262	char(3)	swu, coded
par_r263	char(3)	swu, coded
par_r264	char(3)	swu, coded
par_r265	char(3)	swu, coded
par_r266	char(3)	swu, coded
par_r267	char(3)	swu, coded
par_r268	char(3)	swu, coded
par_r269	char(3)	swu, coded
par_r270	char(3)	swu, coded
par_r271	char(3)	swu, coded
par_r272	char(3)	swu, coded
par_r273	char(3)	swu, coded

par_r274	char(3)	swu, coded
par_r275	char(3)	swu, coded
par_r276	char(3)	swu, coded
par_r277	char(3)	swu, coded
par_r278	char(3)	swu, coded
par_r279	char(3)	swu, coded
par_r280	char(3)	swu, coded
par_prevni	char(3)	swu, coded
par_noci	char(3)	swu, coded
par_noct	char(3)	swu, coded
par_arcf	char(3)	swu, coded
par_flsh	char(3)	
par_flov	char(3)	
par_imdi	char(3)	
par_pric	char(3)	
par_rout	char(3)	
par_voce	char(3)	
par_none	char(3)	
par_caco	char(3)	
par_andy	char(3)	
par_tmusic	char(3)	
par_cstaons	char(3)	

10.2.3.17 FIREMAC

Table 162: FIREMAC

FIREMAC		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
ltg	char(1)	swu, key

ltu	char(2)	swu, key
slot	char(3)	swu, key
func_id	char(2)	
info	char(300)	info
ip_address	char(15)	swu, key
mac_address	char(17)	swu

10.2.3.18 FIREWALL

Table 163: FIREWALL

FIREWALL		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch /Anlage, key
domain	char(8)	domain / Domäne
ltg	char(1)	swu, key
ltu	char(2)	swu, key
slot	char(3)	swu, key
info	char(300)	info
func_id	char(2)	
ip_address	char(15)	swu, key
isdn_access	char(3)	swu, coded
atl_access	char(3)	swu, coded
int_access	char(3)	swu, coded
ext_access	char(3)	swu, coded

10.2.3.19 IPROUTING

Table 164: IPROUTING

IPROUTING		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel

switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
ltg	char(1)	swu, key
ltu	char(2)	swu, key
slot	char(3)	swu, key
func_id	char(2)	func_id
info	char(300)	info
ip_address	char(15)	swu, key
dest_type	char(3)	swu, coded
net_mask	char(15)	swu
router_ip	char(15)	swu

10.2.3.20 NETWORK

Table 165: NETWORK

NETWORK		
:EXPORT:SELECT		
unique_key	int	unique key / eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
ltg	char(1)	swu, key
ltu	char(2)	swu, key
slot	char(3)	swu, key
func_id	char(2)	func_id
info	char(300)	info
ip_address	char(15)	swu
network_name	char(3)	swu, key, coded
net_mask	char(15)	swu
packet_size	char(5)	swu

10.2.3.21 PM_RTG_ACI

Table 166: PM_RTG_ACI

PM_RTG_ACI		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
atcindex	char(1)	swu, key
info	char(300)	info
pm_rout_name	char(30)	swu
access_no	char(22)	swu

10.2.3.22 PM_RTG_FAXO

Table 167: PM_RTG_FAXO

PM_RTG_FAXO		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
info	char(300)	info
overflow_noans	char(3)	swu, coded
name_noans	char(30)	swu
access_no_noans	char(22)	swu
overflow_busy	char(3)	swu, coded
name_busy	char(30)	swu
access_no_busy	char(22)	swu

10.2.3.23 PM_RTG_MWI

Table 168: PM_RTG_MWI

PM_RTG_MWI

:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
info	char(300)	info
msgwaitidx	char(2)	swu, key
pm_rout_name	char(30)	swu
access_no	char(22)	swu
server_type	char(3)	swu, coded

10.2.3.24 REFTA

Table 169: REFTA

REFTA		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
ready_asyn	char(3)	swu, coded
class_of_gen	char(3)	swu
device	char(8)	swu
error_counter	char(5)	swu
board_type	char(8)	swu
pen	char(15)	swu, key
pen_num	int	
pen_num_board	int	
priority	char(2)	swu
supplier	char(3)	swu, coded
block_trunk_ref	char(3)	swu, coded
reftatype	char(3)	swu, key, coded
source_group	char(2)	swu
current_class	char(4)	swu

state	char(16)	state / Zustand, swu

10.2.3.25 REMOTE_TERM

Table 170: REMOTE_TERM

REMOTE_TERM		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
ltg	char(1)	swu, key
ltu	char(2)	swu, key
slot	char(3)	swu, key
func_id	char(2)	finc_id
info	char(300)	info
ip_address	char(15)	swu, key
bit_rate	char(3)	swu, coded
callback	char(3)	swu, coded
no_channels	char(2)	swu
hold_time	char(3)	swu
isdn_station1	char(22)	swu
isdn_station2	char(22)	swu
isdn_station3	char(22)	swu
isdn_station4	char(22)	swu
station1_inc	char(3)	swu, coded
station1_out	char(3)	swu, coded
station2_inc	char(3)	swu, coded
station2_out	char(3)	swu, coded
station3_inc	char(3)	swu, coded
station3_out	char(3)	swu, coded
station4_inc	char(3)	swu, coded
station4_out	char(3)	swu, coded

10.2.3.26 SCREEN_DATA

Table 171: SCREEN_DATA

SCREEN_DATA		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
info	char(300)	info
def_num_op_type	char(3)	swu, coded
def_num_op	char(22)	swu
orig_loc_dest	char(6)	swu
plan_type	char(3)	swu, coded
pre_call_line	char(3)	swu, coded
pre_conn_line	char(3)	swu, coded
call_line	char(3)	swu, coded
conn_line	char(3)	swu, coded
pattern_plan	char(5)	swu
sourcenodenum	char(10)	swu, key

10.2.3.27 SCREEN_PATTERN

Table 172: SCREEN_PATTERN

SCREEN_PATTERN		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain /Domäne
info	char(300)	info
plannum	char(5)	swu, key

10.2.3.28 SCREEN_PATTERN_SUB

Table 173: SCREEN_PATTERN_SUB

SCREEN_PATTERN_SUB		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
pattern	char(22)	swu, key
plannum	char(5)	swu, key

10.2.3.29 TREF_ATTENUATION

Table 174: TREF_ATTENUATION

TREF_ATTENUATION		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
countrycode	char(3)	swu, key, coded
destination	char(3)	swu, key, coded
attenuation	char(3)	swu, coded
origination	char(3)	swu, key, coded
countryvariant	char(3)	swu, coded

10.2.3.30 TREF_TONE

Table 175: TREF_TONE

TREF_TONE		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne

countrycode	char(3)	swu, key, coded
destination	char(3)	swu, key, coded
group	char(3)	swu, key, coded
tone	char(3)	swu, coded
countryvariant	char(3)	swu, coded

10.2.3.31 TRKDEV_BOARD

Table 176: TRKDEV_BOARD

TRKDEV_BOARD		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
trk_dev_type	char(3)	key, coded
board_type	int	key
board_name	char(10)	key
func_id	char(2)	key
part_number	char(12)	part number / Sachnummer, key

10.2.3.32 TRK_ACC_CODE

Table 177: TRK_ACC_CODE

TRK_ACC_CODE		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
info	char(300)	info
accesscode	char(6)	swu, key
cps00	char(3)	swu, coded
cps01	char(3)	swu, coded

cps02	char(3)	swu, coded
cps03	char(3)	swu, coded
cps04	char(3)	swu, coded
cps05	char(3)	swu, coded
cps06	char(3)	swu, coded
cps07	char(3)	swu, coded
cps08	char(3)	swu, coded
cps09	char(3)	swu, coded
cps10	char(3)	swu, coded
cps11	char(3)	swu, coded
cps12	char(3)	swu, coded
cps13	char(3)	swu, coded
cps14	char(3)	swu, coded
cps15	char(3)	swu, coded
call_queue_cap	char(3)	swu
code_call_ac	char(1)	swu
dpln00	char(3)	swu, key, coded
dpln01	char(3)	swu, key, coded
dpln02	char(3)	swu, key, coded
dpln03	char(3)	swu, key, coded
dpln04	char(3)	swu, key, coded
dpln05	char(3)	swu, key, coded
dpln06	char(3)	swu, key, coded
dpln07	char(3)	swu, key, coded
dpln08	char(3)	swu, key, coded
dpln09	char(3)	swu, key, coded
dpln10	char(3)	swu, key, coded
dpln11	char(3)	swu, key, coded
dpln12	char(3)	swu, key, coded
dpln13	char(3)	swu, key, coded
dpln14	char(3)	swu, key, coded
dpln15	char(3)	swu, key, coded

dtmf_conv	char(3)	swu, coded
dtmf_display	char(3)	swu, coded
dtmf_pause_pulse	char(3)	swu, coded
dtmf_txt	char(16)	swu
dest_num	char(3)	swu
dest_node_num	char(11)	swu
phy_node_num	char(11)	swu
character_conv	char(3)	swu, coded
lcr_route_num	char(5)	swu
prev_line_block	char(3)	swu, coded
reroute_cap	char(3)	swu, coded
ref_access_code	char(6)	swu
route_ac_name	char(16)	swu
route_opt_cap	char(3)	swu, coded
no_prec_call_fwd	char(3)	swu, coded
cstaons	char(3)	

10.2.3.33 TRK_ACC_CODE_SUB

Table 178: TRK_ACC_CODE_SUB

TRK_ACC_CODE_SUB		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch 7 Anlage, key
domain	char(8)	domain / Domäne
cps00	char(3)	swu, coded
cps01	char(3)	swu, coded
cps02	char(3)	swu, coded
cps03	char(3)	swu, coded
cps04	char(3)	swu, coded
cps05	char(3)	swu, coded
cps06	char(3)	swu, coded

cps07	char(3)	swu, coded
cps08	char(3)	swu, coded
cps09	char(3)	swu, coded
cps10	char(3)	swu, coded
cps11	char(3)	swu, coded
cps12	char(3)	swu, coded
cps13	char(3)	swu, coded
cps14	char(3)	swu, coded
cps15	char(3)	swu, coded
dpln00	char(3)	swu, key, coded
dpln01	char(3)	swu, key, coded
dpln02	char(3)	swu, key, coded
dpln03	char(3)	swu, key, coded
dpln04	char(3)	swu, key, coded
dpln05	char(3)	swu, key, coded
dpln06	char(3)	swu, key, coded
dpln07	char(3)	swu, key, coded
dpln08	char(3)	swu, key, coded
dpln09	char(3)	swu, key, coded
dpln10	char(3)	swu, key, coded
dpln11	char(3)	swu, key, coded
dpln12	char(3)	swu, key, coded
dpln13	char(3)	swu, key, coded
dpln14	char(3)	swu, key, coded
dpln15	char(3)	swu, key, coded
ref_access_code	char(6)	swu, key
trk_grp_position	char(3)	swu, key
trk_grp_num	char(3)	swu
compat_param	char(3)	swu, coded
svc_fax	char(3)	swu, coded
svc_data	char(3)	swu, coded
svc_voice	char(3)	swu, coded

reserved	char(3)	-

10.2.3.34 TRUNK

Table 179: TRUNK

TRUNK		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
info	char(300)	info
pen	char(15)	swu, key
pen_num	int	pen number
bchannel_group	char(2)	swu, key
trunk_type	char(3)	coded
device_type	char(3)	swu, coded
device	char(3)	swu, coded
trunk_group	char(3)	swu
trunk_name	char(12)	swu
src_grp_idx	char(2)	swu
dnil_clasmrk_g711	char(3)	swu, coded
dnil_clasmrk_ec	char(3)	swu, coded
dnil_clasmrk_g729	char(3)	swu, coded
dnil_clasmrk_g729o	char(3)	
dnil_clasmrk_none	char(3)	swu, coded
dnil_clasmrk_init	char(3)	swu, coded
dnil_clasmrk_vad	char(3)	swu, coded
cot_num	char(3)	swu
cop_num	char(3)	swu
cos_num	char(3)	swu
lcr_cos_voice	char(4)	swu
lcr_cos_data	char(4)	swu

central_byte_attr	char(3)	swu, coded
attenuation_type	char(3)	swu, coded
node_num	char(11)	swu
dest_num	char(3)	swu
network_carrier	char(1)	swu
traffic_zone	char(3)	swu, coded
dpln_group	char(2)	swu
itr_group	char(2)	swu
alarm_num	char(3)	swu
translation	char(3)	swu, coded
did_dig_conv	char(2)	swu
service_rule	char(2)	swu
sidani	char(3)	swu, coded
sicofi_index	char(1)	swu
ind_night_station	char(22)	swu
telcom_cir_id	char(30)	swu
caller_id_dgts	char(3)	swu, coded
digit_pref	char(22)	swu
digit_pref_normal	char(22)	swu
digit_pref_switch	char(22)	swu
digit_pref_house	char(22)	swu
num_sup_digits	char(1)	swu
digits_from_co	char(1)	swu
tbt_index	char(2)	swu
lw_param_block	char(3)	swu
external_auth	char(2)	swu
inc_dial_type	char(3)	swu, coded
inc_dial_var	char(3)	swu
outg_dial_type	char(3)	swu, coded
outg_dial_var	char(3)	swu
mfc_variant	char(2)	swu
trunk_info	char(3)	swu

test_program	char(2)	swu
call_chg_impulse	char(2)	swu
len_of_lin	char(2)	swu
protocol	char(3)	swu, coded
segmentation	char(1)	swu
cot_partner	char(3)	swu
call_forwarding	char(2)	swu
collision_handling	char(3)	swu, coded
congest_controlx	char(3)	swu
congest_controly	char(3)	swu
paging_mode	char(3)	swu, coded
fault_type_index	char(2)	swu
charge_display	char(3)	swu, coded
mult_bearer_chan	char(3)	swu, coded
internat_pref	char(4)	swu
national_pref	char(3)	swu
partner_dom_type	char(3)	swu, coded
partner_dom_num	char(4)	swu
trk_grp_prof_num	char(3)	swu
sup_time_query	char(3)	swu
level2_pref	char(4)	swu
level1_pref	char(3)	swu
access_code_nt	char(6)	swu
satellite_conn	char(3)	swu, coded
transit_counter	char(2)	swu
assumption_idx	char(3)	swu
lw_partner	char(2)	swu
lw_preshaping	char(2)	swu
lw_reserve1	char(2)	swu
lw_reserve2	char(3)	swu
lw_line_type	char(2)	swu
did_hotline_idx	char(5)	swu

dchannel	char(2)	swu
ded_spec_cc	char(4)	swu
bchannel01	char(3)	swu, coded
bchannel02	char(3)	swu, coded
bchannel03	char(3)	swu, coded
bchannel04	char(3)	swu, coded
bchannel05	char(3)	swu, coded
bchannel06	char(3)	swu, coded
bchannel07	char(3)	swu, coded
bchannel08	char(3)	swu, coded
bchannel09	char(3)	swu, coded
bchannel10	char(3)	swu, coded
bchannel11	char(3)	swu, coded
bchannel12	char(3)	swu, coded
bchannel13	char(3)	swu, coded
bchannel14	char(3)	swu, coded
bchannel15	char(3)	swu, coded
bchannel16	char(3)	swu, coded
bchannel17	char(3)	swu, coded
bchannel18	char(3)	swu, coded
bchannel19	char(3)	swu, coded
bchannel20	char(3)	swu, coded
bchannel21	char(3)	swu, coded
bchannel22	char(3)	swu, coded
bchannel23	char(3)	swu, coded
bchannel24	char(3)	swu, coded
bchannel25	char(3)	swu, coded
bchannel26	char(3)	swu, coded
bchannel27	char(3)	swu, coded
bchannel28	char(3)	swu, coded
bchannel29	char(3)	swu, coded
bchannel30	char(3)	swu, coded

bchannel_negot	char(3)	swu, coded
ded_service	char(3)	swu, coded
facility	char(5)	swu
interface_grp	char(3)	swu
interface_id	char(3)	swu
pvc_vpi	char(1)	swu
pvc_vci	char(5)	swu
port_group	char(1)	swu
byte_per_atm_cell	char(2)	swu
error_timer	char(3)	swu
num_rnr_stat	char(3)	swu
fixed_tei_value	char(2)	swu
dest_atm_addr	char(15)	swu
dest_atm_addr_ext	char(3)	swu
num_plan	char(3)	swu, coded
atm_adapt_layer5	char(5)	swu
layer1_active	char(3)	swu, coded
layer2_active	char(3)	swu, coded
layer1_config	char(3)	swu, coded
tei_check	char(3)	swu, coded
master	char(3)	swu, coded
dictation_type	char(1)	swu
announce_type	char(2)	swu
start_dial_timer	char(3)	swu, coded
num_of_wires	char(3)	swu, coded
paging_num_length	char(2)	swu
send_station_num	char(3)	swu, coded
line_alarm_supvis	char(3)	swu, coded
time_ack	char(3)	swu, coded
paging_orders	char(2)	swu
paging_attempts	char(2)	swu
code_caller_norm	char(1)	swu

code_caller_emerg	char(1)	swu
num_char_sent	char(2)	swu
stn_paging_conver	char(3)	swu, coded
wait	char(3)	swu, coded
cancel_order	char(3)	swu, coded
dummy_task_time	char(3)	swu, coded
end_of_dial_time	char(3)	swu, coded
task_time	char(3)	swu, coded
conn_on_answer	char(3)	swu, coded
del_order_on_ans	char(3)	swu, coded
srchblck_end_mrkr	char(3)	swu, coded
srchblck_delimiter	char(3)	swu, coded
pen_standard	char(15)	swu
port_standard	char(15)	swu
mlpp_svc_domain_1	char(3)	swu
mlpp_svc_domain_2	char(3)	swu
mlpp_svc_domain_3	char(3)	swu
mlpp_svc_domain_4	char(3)	swu
mlpp_svc_domain_5	char(3)	swu
mlpp_prc_lvl_fls	char(3)	swu
mlpp_prc_lvl_flso	char(3)	swu
mlpp_prc_lvl_immd	char(3)	swu
mlpp_prc_lvl_sprt	char(3)	swu
mlpp_prc_lvl_dprt	char(3)	swu
mlpp_prc_lvl_prty	char(3)	swu
mlpp_prc_lvl_rtn	char(3)	swu
mlpp_prc_lvl_rtnp	char(3)	swu
dnil_clasmrk_3530	char(3)	swu
dnil_clasmrk_3550	char(3)	swu
dmc_enabled	char(3)	swu
cluster_id	char(4)	swu
dmc_secured	char(3)	

security_level	char(3)	swu, coded
hmusic	char(3)	

10.2.3.35 TRUNK_GRP

Table 180: TRUNK_GRP

TRUNK_GRP		
:EXPORT:SELECT		
unique_key	int	unique key /eindeutiger Schlüssel
switch_name	char(4)	switch / Anlage, key
domain	char(8)	domain / Domäne
info	char(300)	info
acd_call_threshold	char(3)	swu, coded
max_acd_calls	char(4)	swu
max_trunks	char(4)	swu
name	char(18)	swu
priority_class	char(2)	swu
rsv_by_atnd	char(3)	swu, coded
trace_no	char(4)	swu
trunkgroupnum	char(3)	swu, key
dynamic_routing	char(3)	swu, coded
gdtr_rule	char(3)	swu
acd_pm_grp	char(3)	swu
character_conv	char(3)	swu, coded

10.2.4 HIM Objects - Detailed Description

10.2.4.1 BOARD_TYPE

Table 181: BOARD_TYPE

BOARD_TYPE		Board type/Baugruppentyp
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
part_number	char(10)	Part number/Sachnummer, swu
board_name	char(8)	board_name/board_name, swu
board_type	char(3)	board_type/board_type, swu, coded

10.2.4.2 CABINET

Table 182: CABINET

CABINET		Cabinet Data/Schrankdaten
:EXPORT:SELECT		
unique_key	int	unique_key/Schlüssel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Dom"ne
addr	char(4)	Consy address/CONSY-Adresse, swu, key
shelf_no	char(2)	Shelf number/BGR-Nummer, swu, key
phy_addr	char(6)	physical address/physical address, swu
frame	char(8)	frame name/Rahmenname, swu
pid1	char(5)	process id 1/Prozess-ID 1, swu
pid2	char(5)	process id 2/Prozess-ID 2, swu
pid3	char(5)	process id 3/Prozess-ID 3, swu
ltu_no	char(2)	ltu number/LTU-Nummer, swu

10.2.4.3 CABINET_1

Table 183: CABINET_1

CABINET_1		Cabinet Data/Schrankdaten
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
cab	char(8)	cabinet name/Schrank Name, swu
addr	char(4)	Consy address/CONSY-Adresse, swu, key

10.2.4.4 CABINET_VIEW

Table 184: CABINET_VIEW

CABINET_VIEW		Cabinet Data/Schrankdaten
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
addr	char(4)	Consy address/CONSY-Adresse, swu, key
shelf_no	char(2)	Shelf number/BGR-Nummer, swu, key
cab	char(8)	cabinet name/Schrank Name, swu
phy_addr	char(6)	physical address/Physical address, swu
frame	char(8)	frame name/Rahmenname, swu
pid1	char(5)	process id 1/Prozess-ID 1, swu
pid2	char(5)	process id 2/Prozess-ID 2, swu
pid3	char(5)	process id 3/Prozess-ID 3, swu
ltu_no	char(2)	ltu number/LTU-Nummer, swu
part_no	char(14)	Part number/Sachnummer, swu

10.2.4.5 CENTRAL_SIU

Table 185: CENTRAL_SIU

CENTRAL_SIU		Central SIU/
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
slot	char(3)	Slot/EBT, swu
conf_siu	char(14)	configured siu/konfigurierte SIU, swu
is_siu	char(14)	present siu/present siu, swu

10.2.4.6 CENTRAL_SIU_VIEW

Table 186: CENTRAL_SIU_VIEW

CENTRAL_SIU_VIEW		Central SIU view
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
processor	char(3)	Processor/Prozessor, swu, key
slot	char(3)	Slot/EBT, swu
conf_siu	char(14)	configured siu/konfigurierte SIU, swu
is_siu	char(14)	present siu/present siu, swu
lw_no	char(1)	LW number/LW-Nummer, swu
lw_file	char(32)	Loadware file name/Loadware Dateiname, swu
lw_prod	char(20)	Prod. date of loadware/Prod. Datum der Loadware, swu

10.2.4.7 CONSY_CAB

Table 187: CONSY_CAB

CONSY_CAB		Consy cabinet/Consy Schrank
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
part_no	char(14)	Part number/Sachnummer, swu
cab	char(8)	cabinet name/Schrank Name, swu, key

10.2.4.8 CONSY_FRAME

Table 188: CONSY_FRAME

CONSY_FRAME		Consy frame/Consy Rahmen
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
part_no	char(14)	Part number/Sachnummer, swu
frame	char(8)	frame/Rahmen, swu, key

10.2.4.9 COS_DESCRIPTION

Table 189: COS_DESCRIPTION

COS_DESCRIPTION		COS description/COS Beschreibung
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
cos	char(4)	cos/COS, swu
description	char(50)	Description/Beschreibung, swu

10.2.4.10 CUST_FEATURE

Table 190: CUST_FEATURE

CUST_FEATURE		Cust feature
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
fea_name	char(3)	feature name/LM-Name, swu, key, coded
state	char(3)	State/Zustand, swu, coded

10.2.4.11 DBC_CONF_DATA

Table 191: DBC_CONF_DATA

DBC_CONF_DATA		DBC conf data
:EXPORT:SELECT		
mclass	char(12)	System classification/Anlagenklassifikation, swu
mclass2	char(6)	System classification 2/Anlagenklassifikation 2, swu
massemb	char(20)	HW assembly/HW-Assembly, swu
massemb2	char(7)	HW assembly 2/HW-Assembly 2, swu
mdevelop	char(18)	Development line/Entwicklungsleitung, swu
mdevelo2	char(4)	Development line 2/Entwicklungsleitung 2, swu
moperat	char(7)	Operating mode/Betriebsart, swu
mrest	char(5)	Restart type/restart type, swu
mhwarch	char(10)	HW architecture/HW-Architektur, swu
mhwarcht	char(3)	HW architecture type/HW-Architekturtyp, swu
mswitch	char(17)	Switch number/Anlagenummer, swu
mloc	char(8)	location/Standort, swu
mbasisap	char(8)	BAPPL/BAPPL, swu
mdbap	char(8)	DBAPPL/DBAPPL, swu
msystem	char(10)	System ID/System-ID, swu
mltg2	char(2)	LTGs/LTGs, swu
mltu2	char(2)	LTUs/LTUs, swu

DBC_CONF_DATA		DBC conf data
:EXPORT:SELECT		
mline2	char(5)	Log. Lines/Log. Leitungen, swu
mmgsn2	char(3)	MTS BD / GSN/MTS BD / GSN, swu
mlphys2	char(5)	Phys. port/Phys. Port, swu
mpbc2	char(2)	PBCs/PBCs, swu
msiup2	char(2)	SIUs per LTG/SIUs pro LTG, swu
mdiuc2	char(2)	TMD24 per LTU/TMD24 pro LTU, swu
mhwy2	char(3)	HWY / MTS BD/HWY / MTS BD, swu
mhdlc2	char(2)	HDCL / DCL/HDCL / DCL, swu
mpbdc2	char(2)	PBC / DCL/PBC / DCL, swu
msiulin	char(4)	Log. SIU lines/Log. SIU-Leitungen, swu
mconflin	char(4)	Log. conf. lines/Log. Konf.-Leitungen, swu
mdimen	char(8)	DB dimensioning name/DB Dimensionierungsname, swu
mversion	char(3)	Conf. table version/Konf. Tabellenversion, swu

10.2.4.12 DBC_HW_DATA

Table 192: DBC_HW_DATA

DBC_HW_DATA		dbc hardware data
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
mltg2	char(2)	LTGs/LTGs, swu
mltu2	char(2)	LTUs/LTUs, swu
mline2	char(5)	Log. Lines/Log. Leitungen, swu
mmgsn2	char(3)	MTS BD / GSN/MTS BD / GSN, swu
mlphys2	char(5)	Phys. port/Phys. Port, swu
mpbc2	char(2)	PBCs/PBCs, swu
msiup2	char(2)	SIUs per LTG/SIUs pro LTG, swu
mdiuc2	char(2)	TMD24 per LTU/TMD24 pro LTU, swu

DBC_HW_DATA		dbc hardware data
:EXPORT:SELECT		
mhwy2	char(3)	HWY / MTS BD/HWY / MTS BD, swu
mhdlc2	char(2)	HDCL / DCL/HDCL / DCL, swu
mpbdc2	char(2)	PBC / DCL/PBC / DCL, swu
msiulin	char(4)	Log. SIU lines/Log. SIU-Leitungen, swu
mconflin	char(4)	Log. conf. lines/Log. Konf.-Leitungen, swu
mdimen	char(8)	DB dimensioning name/DB Dimensionierungsname, swu
mversion	char(3)	Conf. table version/Konf. Tabellenversion, swu

10.2.4.13 DBC_SYS_DATA

Table 193: DBC_SYS_DATA

DBC_SYS_DATA		dbc system data
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
mclass	char(12)	System classification/Anlagenklassifikation, swu
mclass2	char(6)	System classification 2/Anlagenklassifikation 2, swu
massemb	char(20)	HW assembly/HW-Assembly, swu
massemb2	char(7)	HW assembly 2/HW-Assembly 2, swu
mdevelop	char(18)	Development line/Entwicklungsleitung, swu
mdevelo2	char(4)	Development line 2/Entwicklungsleitung 2, swu
moperat	char(7)	Operating mode/Betriebsart, swu
mrest	char(5)	Restart type/restart type, swu
mhwarch	char(10)	HW architecture/HW-Architektur, swu
mhwarcht	char(3)	HW architecture type/HW-Architekturtyp, swu
mswitch	char(17)	Switch number/Anlagenummer, swu
mloc	char(8)	location/Standort, swu
mbasisap	char(8)	BAPPL/BAPPL, swu
mdbap	char(8)	DBAPPL/DBAPPL, swu

DBC_SYS_DATA		dbc system data
:EXPORT:SELECT		
msystem	char(10)	System ID/System-ID, swu

10.2.4.14 DIMSU_VIEW

Table 194: DIMSU_VIEW

DIMSU_VIEW		dimsu view/
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
dimsu_typ	char(3)	dimsu_typ/dimsu_typ, swu, key, coded
used_dimsu_val	int	used_dimsu_val/used_dimsu_val, swu
max_dimsu_val	int	max_dimsu_val/max_dimsu_val, swu
actual_dimsu_val	int	actual value/actual value, swu
std_dimsu_val	int	standard value/Standardwert, swu
sysmax_dimsu_val	int	System specific max value/systemspezifischer max. Wert, swu

10.2.4.15 FEA_HEADER

Table 195: FEA_HEADER

FEA_HEADER		feature header
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
version	char(4)	Version/Version, swu
serno	char(5)	Serial number/Seriennummer, swu
dongle_hwid	char(8)	Hardware ID/Hardware-ID, swu
entrydate	char(10)	entrydate/entrydate, swu
expiry_date	char(10)	Expiry Date/Ablaufdatum, swu
confirm_code	char(5)	Confirmation Code/Best?tigungscode, swu

FEA_HEADER		feature header
:EXPORT:SELECT		
trial_mode_info	char(3)	Trial Mode Activated/Testmodus aktiviert, swu, coded
trial_remain_days	char(4)	Remaining Days of Trial Mode/Restliche Tage des Testmodus, swu

10.2.4.16 FEA_SOLD

Table 196: FEA_SOLD

FEA_SOLD		sold features/verkaufte Leistungsmerkmale
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
feature	char(3)	feature/Leistungsmerkmale, swu, key, coded
contract	char(5)	sold features/Verkaufte Leistungsmerkmale, swu
used	char(5)	used features/benutzte Leistungsmerkmale, swu
free	char(5)	free features/freie Leistungsmerkmale, swu
marked_for_trial	char(5)	Marked/Blocked for Trial/Markiert/gesperrt für Test, swu

10.2.4.17 FEA_USAGE

Table 197: FEA_USAGE

FEA_USAGE		used features/benutzte Leistungsmerkmale
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
feature	char(3)	feature/Leistungsmerkmale, swu, key, coded
contract	char(3)	sold features/Verkaufte Leistungsmerkmale, swu, coded
used	char(3)	used features/benutzte Leistungsmerkmale, swu, coded

10.2.4.18 HW_INFO

Table 198: HW_INFO

HW_INFO		HW info/Hardware Information
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
hw_arch	char(25)	hardware architecture information/Hardware-Architektur Information, swu
mhwarcht	char(3)	HW architecture type/HW-Architekturtyp, swu
op_mode	char(7)	Operating mode/Betriebsart, swu
swu_proc1	char(8)	SWU processor/SWU Prozessor, swu
swu_mem1	char(6)	Memory on SWU/Speicher am SWU, swu
adp_proc	char(8)	adp processor/ADP Prozessor, swu
adp_mem	char(6)	memory on adp/Speicherplatz auf ADP, swu
sys_mode	char(3)	sys_mode/sys_mode, swu, coded
origin	char(1)	origin/Ursprung
swu_proc2	char(8)	swu/swu
swu_mem2	char(6)	swu/swu

10.2.4.19 LCOS_DESCRIPTION

Table 199: LCOS_DESCRIPTION

LCOS_DESCRIPTION		LCOS description/LCOS Beschreibung
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
lcos	char(4)	lcos/lcos, swu
description	char(50)	Description/Beschreibung, swu

10.2.4.20 LW_ON_CB

Table 200: LW_ON_CB

LW_ON_CB		Loadware CB/Loadware CB
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
assembly	char(8)	assembly/Bestckung, swu, key
pbc_nr	char(2)	Slot number/EBT-Nummer, swu
lw_name	char(32)	name of the loadware/Name der Loadware, swu
lw_date	char(20)	Prod. date of loadware/Prod. Datum der Loadware, swu
origin	char(1)	origin/Ursprung

10.2.4.21 LW_ON_CENTRAL_SIU

Table 201: LW_ON_CENTRAL_SIU

LW_ON_CENTRAL_SIU		Loadware on central siu/Loadware auf central siu
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
lw_no	char(1)	LW number/LW-Nummer, swu, key
lw_file	char(32)	Loadware file name/Loadware Dateiname, swu
lw_prod	char(20)	Prod. date of loadware/Prod. Datum der Loadware, swu
processor	char(3)	Processor/Prozessor, swu
slot	char(3)	Slot/EBT, swu, key

10.2.4.22 LW_ON_PROCESSOR

Table 202: LW_ON_PROCESSOR

LW_ON_PROCESSOR		Loadware on processor/Loadware auf Processor
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
assembly	char(3)	Processor name/Prozessorname, swu, key
info_type	char(8)	information type/Informationstyp, swu, key
lw_id	char(10)	Loadware-ID/Loadware-ID, swu
lw_id_cmp	char(10)	LW ID on CMP/LW-ID am CMP, swu
lw_id_lp	char(10)	LW ID on LP/LW-ID am LP, swu
uw7_mac_adr	char(18)	swu/swu
rmx_mac_adr	char(18)	swu/swu

10.2.4.23 MAC_ADDRESS_VIEW

Table 203: MAC_ADRESS_VIEW

MAC_ADDRESS_VIEW		MAC Address/MAC Adresse
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	key/Anlage,key
domain	char(8)	domain/Domäne
assembly	char(3)	Processor/Prozessor, swu, key
info_type	char(8)	swu/swu, key
uw7_mac_adr	char(18)	C-LAN MAC Address/C-LAN MAC Adresse, swu
rmx_mac_adr	char(18)	IPDA MAC Address/IPDA MAC Adresse, swu

10.2.4.24 PATCH

Table 204: PATCH

PATCH		Patch/Patch
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
patchno	char(12)	Patch number/patch number, swu, key
group_sign	char(11)	group_sign/group_sign, swu
actsign	char(1)	actsign/actsign, swu
group	char(10)	Patch group/Patch-Gruppe, swu
hd	char(1)	patch on hd/Patch auf HD, swu
adp	char(1)	patch on adp/Patch auf ADP, swu
bp	char(2)	patch on bp/patch on bp, swu
optional	char(3)	Optional patch/Optional Patch, swu, coded

10.2.4.25 PEN_LW

Table 205: PEN_LW

PEN_LW		Pen LW/Pen LW
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
lw_no	char(1)	LW number/LW-Nummer, swu, key
lw_inter_ver	char(2)	LW interface version/LW-Schnittstellenversion, swu
lw_name	char(32)	name of the loadware/Name der Loadware, swu
lw_date	char(20)	Prod. date of loadware/Prod. Datum der Loadware, swu
pen	char(9)	PEN/pen, swu, key
ltu	char(2)	ltu/ltu, swu
ebt	char(3)	Slot/EBT, swu

10.2.4.26 PORT_STATION_VIEW

Table 206: PORT_STATION_VIEW

PORT_STATION_VIEW		Port station view
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
pen	char(9)	PEN/pen, swu, key
circuit	char(3)	circuit number/Satznummer, swu, key
extension	char(8)	Extension/Nebenstelle, swu
name	char(30)	Name/Name, swu
devtyp	char(10)	devtyp/devtyp, swu
stbnum	char(3)	stbnum/stbnum, swu
description	char(50)	Description/Beschreibung, swu

10.2.4.27 PROJ_PLAN_DATA

Table 207: PROJ_PLAN_DATA

PROJ_PLAN_DATA		Project plan data
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
create_date	char(10)	create_date/create_date, swu
create_time	char(8)	create_time/create_time, swu

10.2.4.28 PSIO_VIEW

Table 208: PSIO_VIEW

PSIO_VIEW		PSIO view
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel

PSIO_VIEW		PSIO view
:EXPORT:SELECT		
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
pen	char(9)	PEN/pen, swu, key
is_card	char(10)	is_card/is_card, swu
firmware	char(6)	firmware on the card/Firmware auf die BG, swu
type	char(8)	type/Art, swu

10.2.4.29 REPLACED_AMO

Table 209: REPLACED_AMO

REPLACED_AMO		Replaced AMO
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
prog_sys	char(8)	Program system/Programm-System, swu, key
amo_name	char(6)	AMO Name/AMO Name, swu, key
is_amo	char(22)	is amo in prog. system/IS AMO im Prog. System, swu
should_amo	char(22)	Should AMO in prog. system/Soll-AMO im Programmsystem, swu

10.2.4.30 SCSI_DEV

Table 210: SCSI_DEV

SCSI_DEV		SCSI Device
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
dev_type	char(13)	device type/Ger?tetyp, swu
scsi_id	char(1)	scsi id/SCSI-ID, swu

SCSI_DEV		SCSI Device
:EXPORT:SELECT		
dev_name	char(25)	device name/Gerätename, swu
fw	char(4)	firmware/firmware, swu
load_dev	char(3)	load_dev/load_dev, swu, coded
contr_no	char(1)	contr_no/contr_no, swu, key
vendor	char(8)	Vendor name/Lieferantname, swu
product	char(16)	Product name/Produktname, swu
size	char(8)	size/Grösse, swu

10.2.4.31 SCSI_DEV_VIEW

Table 211: SCSI_DEV_VIEW

SCSI_DEV_VIEW		SCSI device view
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
dev_type	char(13)	device type/Gerätetyp, swu
scsi_id	char(1)	scsi id/SCSI-ID, swu, key
dev_name	char(25)	device name/Gerätename, swu
fw	char(4)	firmware/firmware, swu
load_dev	char(3)	load_dev/load_dev, swu, coded

10.2.4.32 SERIAL_LINE

Table 212: SERIAL_LINE

SERIAL_LINE		Serial line/Serielle Leitung
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne

SERIAL_LINE		Serial line/Serielle Leitung
:EXPORT:SELECT		
card	char(8)	board/Baugruppe, swu
channel	char(2)	line number/Leitungsnummer, swu, key
baud	char(5)	Baud Rate/Baudrate, swu
dev_name	char(8)	device name/Gerätename, swu
dev_type	char(18)	device type/Gerätetyp, swu
line_type	char(3)	Line Type/Leitungstyp, swu
type_no	char(1)	type_no/type_no, swu
lcxno	char(1)	lcxno/lcxno, swu

10.2.4.33 SHELF

Table 213: SHELF

SHELF		Schelf/Baugruppe
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
addr	char(9)	shelf address/Baugruppenadresse, swu, key
ltuc_code_no	char(10)	ltuc_code_no/ltuc_code_no, swu
ltu_type	char(10)	ltu_type/ltu_type, swu
phy_addr	char(7)	physical address/physical address, swu
frame	char(8)	frame/Rahmen, swu
ltu	char(2)	ltu/ltu, swu, key
overlay	char(2)	overlay/overlay, swu
location	char(48)	location/Standort, swu
network_type	char(3)	swu/swu, coded
network_addr	char(15)	swu/swu
remote	char(3)	coded/coded
act_bchlcnt	char(3)	swu/swu
max_bchlcnt	char(3)	swu/swu

SHELF		Schelf/Baugruppe
:EXPORT:SELECT		
ip_address	char(15)	swu/swu
src_grp	char(2)	swu/swu
ape_control	char(3)	swu/swu, coded
ape_group	char(2)	swu/swu

10.2.4.34 SHELF VIEW

Table 214: SHELF_VIEW

SHELF_VIEW		Schelf view/Baugruppen view
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
addr	char(9)	shelf address/Baugruppenadresse, swu, key
ltuc_code_no	char(10)	ltuc_code_no/ltuc_code_no, swu
ltu_type	char(10)	ltu_type/ltu_type, swu
part_no	char(14)	Part number/Sachnummer, swu
phy_addr	char(7)	physical address/physical address, swu
location	char(48)	location/Standort, swu
network_type	char(3)	swu/swu, coded
network_addr	char(15)	swu/swu
remote	char(3)	coded/coded

10.2.4.35 SWITCH_APS

Table 215: SWITCH_APS

SWITCH_APS		Switch APS
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key

SWITCH_APS		Switch APS
:EXPORT:SELECT		
domain	char(8)	domain/Domäne
prog_sys	char(8)	Program system/Programm-System, swu, key
version	char(2)	Version/Version, swu
corr_ver	char(3)	correction version/Kurrekturversion, swu
partno	char(17)	Part number/Sachnummer, swu

10.2.4.36 SWITCH_NO

Table 216: SWITCH_NO

SWITCH_NO		Switch number/Anlagennummer
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
switch_number	char(17)	switch_number/Anagennummer, swu, key

10.2.4.37 SWU_BOARD

Table 217: SWU_BOARD

SWU_BOARD		SWU Board/SWU Baugruppe
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
pen	char(9)	PEN/pen, swu
ltg	char(2)	ltg/ltg, swu, key
ltu	char(2)	ltu/ltu, swu, key
slot	char(3)	Slot/EBT, swu, key
conf_card	char(10)	configured card/konfigurierte Baugruppe, swu
type	char(8)	board type/board type, swu

SWU_BOARD		SWU Board/SWU Baugruppe
:EXPORT:SELECT		
fid	char(3)	functional id/Funktions-ID, swu
is_card	char(10)	Actual card/aktuelle BG, swu
state	char(2)	State of the card/Zustand der BG, swu
firmware	char(6)	firmware on the card/Firmware auf die BG, swu
mode	char(8)	mode/mode, swu
btype	char(3)	board type/board type, swu, coded
overlay	char(2)	overlay/overlay, swu
mac_address	char(17)	
tdmlaw	char(6)	
ui_mode	char(12)	

10.2.4.38 SWU_BOARD_VIEW

Table 218: SWU_BOARD_VIEW

SWU_BOARD_VIEW		SWU Board view
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
pen	char(9)	PEN/pen, swu, key
conf_card	char(10)	configured card/konfigurierte Baugruppe, swu
type	char(8)	board type/board type, swu
fid	char(3)	functional id/Funktions-ID, swu
is_card	char(10)	Actual card/aktuelle BG, swu
state	char(2)	State of the card/Zustand der BG, swu
firmware	char(5)	firmware on the card/Firmware auf die BG, swu
mode	char(8)	mode/mode, swu
btype	char(3)	board type/board type, swu, coded
lw_no	char(1)	LW number/LW-Nummer, swu, key
lw_inter_ver	char(2)	LW interface version/LW-Schnittstellenversion, swu

SWU_BOARD_VIEW		SWU Board view
:EXPORT:SELECT		
lw_name	char(32)	name of the loadware/Name der Loadware, swu
lw_date	char(20)	Prod. date of loadware/Prod. Datum der Loadware, swu
overlay	char(2)	overlay/overlay, swu
mac_address	char(17)	
tdmlaw	char(6)	
ui_mode	char(12)	

10.2.4.39 SWU_N_DATA

Table 219: SWU_N_DATA

SWU_N_DATA		SWU
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
info_nr	int	information number/Informationsnummer, swu, key
creation_date	char(10)	creation_date/creation_date, swu
text	char(77)	Text/Text, swu

10.2.4.40 SWU_PORT_ATTENDANT

Table 220: SWU_PORT_ATTENDANT

SWU_PORT_ATTENDANT		SWU port attendant
:EXPORT:SELECT		
unique_key	int	unique_key/Schlssel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
att_pen	char(9)	Attendant Pen/VPL Lage, swu, key
att_port	char(3)	Attendant Port/VPL Port, swu, key
att_extension	char(12)	Attendant Extension/VPL Nebenstelle, swu

SWU_PORT_ATTENDANT		SWU port attendant
:EXPORT:SELECT		
at_grp_num	char(3)	Attendant Group Number/VPL-Gruppennummer, swu
att_language	char(3)	Attendant Language/VPL Sprache, swu, coded
fpen	char(13)	fpen/fpen, swu

10.2.4.41 SWU_PORT_SPECIAL

Table 221: SWU_PORT_SPECIAL

SWU_PORT_SPECIAL		SWU port special
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
pen	char(9)	PEN/pen, swu, key
port	char(3)	Port/Port, swu, key
type	char(6)	type/Art, swu
number	char(6)	Number/Anzahl, swu
fpen	char(13)	fpen/fpen, swu

10.2.4.42 SWU_T_DATA

Table 222: SWU_T_DATA

SWU_T_DATA		SWU T data
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
info_nr	char(7)	information number/Informationsnummer, swu, key
creation_date	char(10)	creation_date/creation_date, swu
text	char(40)	Text/Text, swu
quantity	char(5)	Quantity/Anzahl, swu

SWU_T_DATA		SWU T data
:EXPORT:SELECT		
text_2	char(10)	Additional text/zusätzlicher Trext, swu

10.2.4.43 SW_VERSION

Table 223: SW_VERSION

SW_VERSION		SW version
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
prog_on	char(3)	prog_on/prog_on, swu, key
partno	char(17)	Part number/Sachnummer, swu
idx_code	char(8)	idx_code/idx_code, swu
version	char(2)	Version/Version, swu
sys_rel	char(2)	System release/System-Release, swu
originator	char(1)	Originator/Ersteller, swu
corr_ver	char(3)	correction version/Kurrekturversion, swu
rev_no	char(2)	Revision number/Revision-Nummer, swu

10.2.4.44 TRUNK_VIEW

Table 224: TRUNK_VIEW

TRUNK_VIEW		Trunk view
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
pen	char(9)	PEN/pen, swu, key
port	char(3)	Port/Port, swu, key
device	char(3)	device/Ger?t, swu, coded

TRUNK_VIEW		Trunk view
:EXPORT:SELECT		
protocol	char(3)	Protocol/Protokoll, swu, coded
dest_num	char(3)	dest_num/dest_num, swu
trunk_group	char(3)	trunk_group/trunk_group, swu
trunk_name	char(12)	trunk_name/trunk_name, swu

10.2.4.45 UNIX_SW

Table 225: UNIX_SW

UNIX_SW		Unix software
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
pkg_name	char(50)	Unix package name/UNIX Paketname, swu, key
version	char(20)	Unix SW version/Unix SW Version, swu
inst_date	char(40)	installation date/Installationsdatum, swu
inst_state	char(3)	installation state/Installationszustand, swu, coded

10.2.4.46 WELCOME_PAGE

Table 226: WELCOME_PAGE

WELCOME_PAGE_VIEW		welcome page/Startseite
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
switch_number	char(17)	switch_number/switch_number, swu
version	char(4)	Version/Version, swu
sys_rel	char(2)	System release/System-Release, swu
rel_no	char(2)	rel_no/rel_no, swu

WELCOME_PAGE_VIEW		welcome page/Startseite
:EXPORT:SELECT		
rev_no	char(2)	swu/swu
hw_arch	char(25)	hardware architecture information/Hardware-Architektur Information, swu
hw_arch_type	char(3)	hw_arch_type/hw_arch_type, swu
op_mode	char(7)	Operating mode/Betriebsart, swu
swu_proc1	char(8)	SWU processor/SWU Prozessor, swu
swu_mem1	char(6)	Memory on SWU/Speicher am SWU, swu
sys_mode	char(3)	sys_mode/sys_mode, swu, coded
swu_proc2	char(8)	swu/swu
swu_mem2	char(6)	swu/swu

10.2.4.47 Y_APS

Table 227: Y_APS

Y_APS		Systeminfo/Systeminformationen
:EXPORT:SELECT		
unique_key	int	unique_key/Schluessel
switch_name	char(4)	switch_name/Anlage, key
domain	char(8)	domain/Domäne
prog_on	char(3)	Program on medium/Programm am medium, swu, key
partno	char(17)	Part number/Sachnummer, swu
idx_code	char(8)	idx_code/idx_code, swu
version	char(2)	Version/Version, swu
sys_rel	char(2)	System release/System-Release, swu
rel_no	char(2)	rel_no/rel_no, swu
originator	char(1)	Originator/Ersteller, swu
corr_ver	char(3)	correction version/Kurrekturversion, swu
rev_no	char(2)	Revision number/Revision-Nummer, swu
prog_sys	char(6)	prog_sys/prog_sys, swu
origin	char(1)	origin/Ursprung

10.2.5 Accounting Management Objects - Detailed Description

NOTICE: Following tables contain ODFs common for both, OpenScape 4000 Manager and OpenScape 4000 Assistant. In case that the field is Manager (or Assistant) specific, it is stated in the third column of the table.

10.2.5.1 AM_VNR_PERS

Table 228: AM_VNR_PERS

AM_VNR_PERS		Personal (subscriber) data with VNR extensions
:EXPORT:SELECT:SELECT		
identifier	int	unique_key, key
country	char(2)	country
company	char(30)	Company
org1	char(40)	Org 1
org2	char(40)	Org 2
org3	char(40)	Org 3
org4	char(40)	Org 4
org5	char(40)	Org 5
org6	char(40)	Org 6
name	char(50)	Name
sortname	char(60)	Sortname
christianname	char(32)	First Name
title	char(15)	Title
salutation	char(15)	Salutation
location	char(15)	Location
building	char(30)	Building
longname	char(15)	Longname
extension	char(12)	Extension
switch	char(4)	switch_name
position	int	sequ_num
tienum	char(6)	tienum
ext_areacode	char(12)	ext_areacode

AM_VNR_PERS		Personal (subscriber) data with VNR extensions
:EXPORT:SELECT:SELECT		
ext_netcode	char(16)	ext_netcode
ext_ext	char(12)	id
faxnumber	char(22)	faxnumber
import_dat	char(1)	import_dat
owner_group	char(30)	owner_group
modify_date	char(14)	Modification date
access_control	char(1)	access_control
text1	char(50)	Text1
text2	char(50)	Text2
text3	char(50)	Text3
direct_trunk_acc	char(1)	Direct CO Trunk Line
equipment	char(160)	Device Configuration
l1	char(4)	Service Type 1
l2	char(4)	Service Type 2
l3	char(4)	Service Type 3
l4	char(4)	Service Type 4
l5	char(4)	Service Type 5
l6	char(4)	Service Type 6
l7	char(4)	Service Type 7
l8	char(4)	Service Type 8
l9	char(4)	Service Type 9
l10	char(4)	Service Type 10
l11	char(4)	Service Type 11
l12	char(4)	Service Type 12
l13	char(4)	Service Type 13
l14	char(4)	Service Type 14
l15	char(4)	Service Type 15
l16	char(4)	Service Type 16
l17	char(4)	Service Type 17
l18	char(4)	Service Type 18

AM_VNR_PERS		Personal (subscriber) data with VNR extensions
:EXPORT:SELECT:SELECT		
l19	char(4)	Service Type 19
l20	char(4)	Service Type 20
l_prefix	char(7)	Service type prefix
text4	char(50)	Keywords4
ext_countrycode	char(4)	Ext. Country Code
chargeid	char(32)	Basic Charge Class
basic_charge_class	char(12)	Basic Charge Class
bankcode	char(15)	Bank Sort Code
accountnum	char(15)	Account Number
xpr_exists	char(3)	Mailbox, coded
xpr_key	char(25)	Mailbox ID
xpr_pin	char(12)	pin_num
xpr_server	char(25)	XPR Server
xpr_vmsadmin	char(3)	Administrator, coded
xpr_broadcast	char(3)	Create broadcast messages, coded
xpr_dct_send	char(3)	Create dictation, coded
xpr_dct_receive	char(3)	Receive dictation, coded
xpr_password	char(3)	Password required, coded
xpr_name	char(3)	Name announcement required, coded
xpr_flag1	char(3)	No Company Announcement, coded
xpr_flag2	char(3)	Prompt 'Infobox reached', coded
xpr_connect	char(3)	Connection Rights, coded
xpr_quota	char(12)	Quota
xpr_reserve1	char(12)	Delegate
xpr_reserve2	char(100)	Customer-specific 4
persnum	char(32)	persnum
domain	char(8)	domain
idf	char(10)	IDF
idf_cable_pair	char(10)	Cable pair
info_5	char(50)	Info 5

AM_VNR_PERS		Personal (subscriber) data with VNR extensions
:EXPORT:SELECT:SELECT		
info_6	char(50)	Info 6
info_7	char(50)	Info 7
info_8	char(50)	Info 8
info_9	char(50)	Info 9
info_10	char(50)	Info 10
info_11	char(50)	Info 11
info_12	char(50)	Info 12
info_13	char(50)	Info 13
info_14	char(50)	Info 14
info_15	char(50)	Info 15
info_16	char(50)	Info 16
info_17	char(50)	info13
info_18	char(50)	info14
info_19	char(50)	info15
v_switch	char(4)	v_switch
usid	char(40)	usid
umuid	char(50)	umuid
um_user	char(3)	um_user, coded
um_marked	char(3)	um_marked, coded
um_operation_id	char(50)	um_operation_id
ape_group_no	char(2)	ape_group_no
xpr_delpwd	char(3)	xpr_delpwd, coded
xpr_delpwd_tmp	char(3)	xpr_delpwd_tmp, coded
tenant_group	char(20)	Note: **Manager only**
e_mail_addr	char(320)	
switch_unique_key	int	unique_key
switch_typ	char(15)	System type
version	char(5)	Version
info	char(300)	Info
switch_country	char(2)	Country

AM_VNR_PERS		Personal (subscriber) data with VNR extensions
:EXPORT:SELECT:SELECT		
areacode	char(16)	id
netcode	char(16)	ext_areacode
upload	char(1)	upload
amo_lang	char(3)	AMO Language, coded
sys_verify	int	sys_verify
nodenum	char(10)	nodenum
version_old	char(5)	version_old
last_delta	char(19)	Last upload delta
last_all	char(19)	Last upload all
upload_cnt	int	Upload counter
upload_day	char(3)	PBX data, coded
exp_upl_date	char(19)	Next delta
spdl_acc_cod1	char(6)	Access code for speed dial list 1
spdl_acc_cod2	char(6)	Access code for speed dial list 2
admin	char(3)	PBX data, coded
vms_admin	char(3)	VMS data, coded
vms_server	char(3)	VM server, coded
vms_switch	char(4)	VMS switch
vms_last_delta	char(19)	Last VMS delta
vms_last_all	char(19)	Last VMS all
vms_sys_verify	int	vms_sys_verify
vms_upload	char(1)	VMS upload
vms_upload_cnt	int	VMS upload counter
vms_dbsync	char(1)	vms_dbsync
switch_dbsync	char(1)	switch_dbsync
cmi_button_std	char(3)	CMI key layout
cmi_cos	char(4)	CMI cos
cmi_lcos	char(4)	CMI-LCOS
lcr_data	char(3)	LCR Support, coded
lcr_last_all	char(19)	Last LCR upload all

AM_VNR_PERS		Personal (subscriber) data with VNR extensions
:EXPORT:SELECT:SELECT		
lcr_last_delta	char(19)	Last LCR upload delta
lcr_upload_cnt	int	Upload counter for LCR
lcr_upload_day	char(3)	LCR Data, coded
lcr_dbsync	char(1)	Upload flag for LCR
lcr_upload	char(1)	Upload status
lcr_admin	char(3)	LCR Data, coded
delta_ctrl	char(3)	Request upload data via the DMS, coded
sec_node	char(10)	Extended Node Number
language1	char(3)	Language 1, coded
language2	char(3)	Language 2, coded
language3	char(3)	Language 3, coded
language4	char(3)	Language 4, coded
language5	char(3)	Language 5, coded
countrycode	char(4)	Country Table
uplo2_aps	char(13)	UPLO2 APS
uplol_aps	char(13)	UPLOL APS
uplvm_aps	char(13)	UPLVM APS
knotenkz	char(22)	Private code
consistent	char(3)	consistent, coded
net_team	char(3)	Exec./Secr. Team Administration, coded
dtb_server	char(20)	DTB_Server
upload_txt	char(3)	upload status, coded
vms_upload_txt	char(3)	vms upload status, coded
lcr_upload_txt	char(3)	lcr upload status, coded
switch_number	char(17)	switch_number
manual_plen	char(2)	Manual PIN Length
passwd_plen	char(2)	Authorization PIN Length
pref_route_index	char(4)	Preferred Route Index
preliminary	char(3)	Preliminary, coded
preliminary_info	int	preliminary_info

AM_VNR_PERS		Personal (subscriber) data with VNR extensions
:EXPORT:SELECT:SELECT		
account_length	char(2)	Account Length
dss_partners	char(2)	dss_partners
sysm_flag	char(3)	sysm_flag, coded
no_name	char(16)	no_name
pbx_upload	char(1)	pbx_upload
not_upload	char(1)	not_upload
not_dbsync	char(1)	not_dbsync
not_last_all	char(19)	not_last_all
backup_swu	char(1)	backup_swu
upload_file	char(1)	upload_file
upload_him_swu	char(1)	upload_him_swu
upload_him_adp	char(1)	upload_him_adp
him_dbsync	char(1)	him_dbsync
him_last_all	char(19)	him_last_all
startupd_time	char(5)	startupd_time
delta_time	char(3)	delta_time
pbx_upload_txt	char(3)	pbx_upload_txt, coded
not_upload_txt	char(3)	not_upload_txt, coded
upload_file_txt	char(3)	upload_file_txt, coded
upl_him_swu_txt	char(3)	upl_him_swu_txt, coded
upl_him_adp_txt	char(3)	upl_him_adp_txt, coded
hicom_sw_version	char(13)	hicom_sw_version
epnp_barrier	char(6)	epnp_barrier
subdomain	char(50)	Subdomain
transit_switch	char(3)	transit_switch, coded
transit_index	char(4)	transit_index
lcr_exp_upl_date	char(19)	lcr_exp_upl_date
legk_support	char(3)	legk_support, coded
dbsync_lock	char(3)	dbsync_lock, coded
cmi_max_subsc	char(5)	

AM_VNR_PERS		Personal (subscriber) data with VNR extensions
:EXPORT:SELECT:SELECT		
pbx_upl_hour_adp	char(2)	
pbx_upl_min_adp	char(2)	
lcr_upl_hour_adp	char(2)	
lcr_upl_min_adp	char(2)	
pbx_upl_hour_mgr	char(2)	
pbx_upl_min_mgr	char(2)	
lcr_upl_hour_mgr	char(2)	
lcr_upl_min_mgr	char(2)	
cmi_cos2	char(4)	
cmi_lcos2	char(4)	
cmi_lcosd	char(4)	
cmi_lcosd2	char(4)	
wabegrpnum	char(2)	
vbzgrpnum	char(2)	
nunp_active	char(3)	coded
spe_support	char(3)	swu, coded
sec_tdm_subsr	char(3)	swu, coded
sec_tdm_trunk	char(3)	swu, coded
sec_ip_subsr	char(3)	swu, coded
sec_ip_trunk	char(3)	swu, coded
dns_ip_address	char(15)	swu
dls_ip_address	char(15)	swu
dls_port	char(5)	swu
key_len	char(3)	swu, coded
mikey_opt	char(3)	swu, coded
dls_accept_pass	char(3)	swu, coded
vnrad_active	char(3)	coded. Note: **Manager only**
vnr_switch_name	char(4)	Note: **Manager only**
seclvdsp	char(3)	swu, coded
upl_time_err	char(3)	coded. Note: **Manager only**

AM_VNR_PERS		Personal (subscriber) data with VNR extensions
:EXPORT:SELECT:SELECT		
display_mode	char(3)	swu, coded
numbering_plan_unique_key	int	unique_key
virtual_node_id	char(10)	virtual_node_id, swu, key
partner	char(3)	Partner, swu, coded
default_entry	char(3)	default_entry, swu, coded
isdn_cc	char(4)	isdn_cc, swu
isdn_ac	char(6)	isdn_ac, swu
isdn_lc	char(10)	isdn_lc, swu
isdn_ul	char(3)	isdn_ul, swu, coded
isdn_skip_digits	char(2)	isdn_skip_digits, swu
pnp_l2	char(4)	pnp_l2, swu
pnp_l1	char(6)	pnp_l1, swu
pnp_l0	char(10)	pnp_l0, swu
pnp_ul	char(3)	pnp_ul, swu, coded
pnp_skip_digits	char(2)	pnp_skip_digits, swu
virtual_node_num	char(6)	virtual_node_num, swu
skip_digits	char(1)	skip_digits, swu
cac_prefix	char(3)	cac_prefix, swu
numbering_plan_info	char(300)	info
sync_by	char(4)	
sync_time	int	
default_wabe	char(2)	swu
wabe0	char(3)	swu, coded
wabe1	char(3)	swu, coded
wabe2	char(3)	swu, coded
wabe3	char(3)	swu, coded
wabe4	char(3)	swu, coded
wabe5	char(3)	swu, coded
wabe6	char(3)	swu, coded
wabe7	char(3)	swu, coded

AM_VNR_PERS		Personal (subscriber) data with VNR extensions
:EXPORT:SELECT:SELECT		
wabe8	char(3)	swu, coded
wabe9	char(3)	swu, coded
wabe10	char(3)	swu, coded
wabe11	char(3)	swu, coded
wabe12	char(3)	swu, coded
wabe13	char(3)	swu, coded
wabe14	char(3)	swu, coded
wabe15	char(3)	swu, coded
E164	char(18)	

10.2.5.2 AM_VNR_PIN

Table 229: AM_VNR_PIN

AM_VNR_PIN		PIN data with VNR extensions
:EXPORT:SELECT:SELECT		
identifier	int	
domain	char(8)	
extension	char(12)	
switch	char(4) char(10)	Note**: Value for Manager** Note: **Value for Assistant**
switch_original	char(4)	Note: **Assistant only**
pin_num	char(12)	
pin_type	char(3)	
position	int	
pin_class	char(2)	
pinindiv	char(3)	
numbering_plan_unique_key	int	unique_key
virtual_node_id	char(10)	virtual_node_id, swu, key
partner	char(3)	Partner, swu, coded
default_entry	char(3)	default_entry, swu, coded
isdn_cc	char(4)	isdn_cc, swu

AM_VNR_PIN		PIN data with VNR extensions
:EXPORT:SELECT:SELECT		
isdn_ac	char(6)	isdn_ac, swu
isdn_lc	char(10)	isdn_lc, swu
isdn_ul	char(3)	isdn_ul, swu, coded
isdn_skip_digits	char(2)	isdn_skip_digits, swu
pnp_l2	char(4)	pnp_l2, swu
pnp_l1	char(6)	pnp_l1, swu
pnp_l0	char(10)	pnp_l0, swu
pnp_ul	char(3)	pnp_ul, swu, coded
pnp_skip_digits	char(2)	pnp_skip_digits, swu
virtual_node_num	char(6)	virtual_node_num, swu
skip_digits	char(1)	skip_digits, swu
cac_prefix	char(3)	cac_prefix, swu
numbering_plan_info	char(300)	info
sync_by	char(4)	
sync_time	int	
default_wabe	char(2)	swu
wabe0	char(3)	swu, coded
wabe1	char(3)	swu, coded
wabe2	char(3)	swu, coded
wabe3	char(3)	swu, coded
wabe4	char(3)	swu, coded
wabe5	char(3)	swu, coded
wabe6	char(3)	swu, coded
wabe7	char(3)	swu, coded
wabe8	char(3)	swu, coded
wabe9	char(3)	swu, coded
wabe10	char(3)	swu, coded
wabe11	char(3)	swu, coded
wabe12	char(3)	swu, coded
wabe13	char(3)	swu, coded

AM_VNR_PIN		PIN data with VNR extensions
:EXPORT:SELECT:SELECT		
wabe14	char(3)	swu, coded
wabe15	char(3)	swu, coded
E164	char(18)	

10.2.5.3 AM_VNR_SWITCHDATA

Table 230: AM_VNR_SWITCHDATA

AM_VNR_SWITCHDATA		Switch data with VNR extensions
:EXPORT:SELECT		
unique_key	int	unique_key
switch_name_original	char(4)	switch_name. Note: **Assistant only**
switch_name	char(10) char(4)	virtual_node_id, key. Note: **Assistant only** switch_name, key. Note: **Manager only**
switch_typ	char(15)	System type
version	char(5)	Version
info	char(300)	info
country	char(2)	country
areacode	char(16)	id
netcode	char(16)	ext_areacode
upload	char(1)	upload
amo_lang	char(3)	AMO Language, coded
sys_verify	int	sys_verify
nodenum	char(10)	nodenum
version_old	char(5)	version_old
last_delta	char(19)	Last upload delta
last_all	char(19)	Last upload all
domain	char(8)	domain
upload_cnt	int	Upload counter
upload_day	char(3)	PBX data, coded
exp_upl_date	char(19)	Next delta

AM_VNR_SWITCHDATA		Switch data with VNR extensions
:EXPORT:SELECT		
spdl_acc_cod1	char(6)	Access code for speed dial list 1
spdl_acc_cod2	char(6)	Access code for speed dial list 2
admin	char(3)	PBX data, coded
vms_admin	char(3)	VMS data, coded
vms_server	char(3)	VM server, coded
vms_switch	char(4)	VMS switch
vms_last_delta	char(19)	Last VMS delta
vms_last_all	char(19)	Last VMS all
vms_sys_verify	int	vms_sys_verify
vms_upload	char(1)	VMS upload
vms_upload_cnt	int	VMS upload counter
vms_dbsync	char(1)	vms_dbsync
switch_dbsync	char(1)	switch_dbsync
cmi_button_std	char(3)	CMI key layout
cmi_cos	char(4)	CMI cos
cmi_lcos	char(4)	CMI-LCOS
lcr_data	char(3)	LCR Support, coded
lcr_last_all	char(19)	Last LCR upload all
lcr_last_delta	char(19)	Last LCR upload delta
lcr_upload_cnt	int	Upload counter for LCR
lcr_upload_day	char(3)	LCR Data, coded
lcr_dbsync	char(1)	Upload flag for LCR
lcr_upload	char(1)	Upload status
lcr_admin	char(3)	LCR Data, coded
delta_ctrl	char(3)	Request upload data via the DMS, coded
sec_node	char(10)	Extended Node Number
language1	char(3)	Language 1, coded
language2	char(3)	Language 2, coded
language3	char(3)	Language 3, coded
language4	char(3)	Language 4, coded

AM_VNR_SWITCHDATA		Switch data with VNR extensions
:EXPORT:SELECT		
language5	char(3)	Language 5, coded
countrycode	char(4)	Country Table
uplo2_aps	char(13)	UPLO2 APS
uplol_aps	char(13)	UPLOL APS
uplvm_aps	char(13)	UPLVM APS
knotenkz	char(22)	Private code
consistent	char(3)	consistent, coded
net_team	char(3)	Exec./Secr. Team Administration, coded
dtb_server	char(20)	DTB_Server
upload_txt	char(3)	upload status, coded
vms_upload_txt	char(3)	vms upload status, coded
lcr_upload_txt	char(3)	lcr upload status, coded
switch_number	char(17)	switch_number
manual_plen	char(2)	Manual PIN Length
passwd_plen	char(2)	Authorization PIN Length
pref_route_index	char(3)	Preferred Route Index
preliminary	char(4)	Preliminary, coded
preliminary_info	int	preliminary_info
account_length	char(2)	Account Length
dss_partners	char(2)	dss_partners
sysm_flag	char(3)	sysm_flag, coded
no_name	char(16)	no_name
pbx_upload	char(1)	pbx_upload
not_upload	char(1)	not_upload
not_dbsync	char(1)	not_dbsync
not_last_all	char(19)	not_last_all
backup_swu	char(1)	backup_swu
upload_file	char(1)	upload_file
upload_him_swu	char(1)	upload_him_swu
upload_him_adp	char(1)	upload_him_adp

AM_VNR_SWITCHDATA		Switch data with VNR extensions
:EXPORT:SELECT		
him_dbsync	char(1)	him_dbsync
him_last_all	char(19)	him_last_all
startupd_time	char(5)	startupd_time
delta_time	char(3)	delta_time
pbx_upload_txt	char(3)	pbx_upload_txt, coded
not_upload_txt	char(3)	not_upload_txt, coded
upload_file_txt	char(3)	upload_file_txt, coded
upl_him_swu_txt	char(3)	upl_him_swu_txt, coded
upl_him_adp_txt	char(3)	upl_him_adp_txt, coded
hicom_sw_version	char(13)	hicom_sw_version
epnp_barrier	char(6)	epnp_barrier
subdomain	char(50)	Subdomain
transit_switch	char(3)	transit_switch, coded
transit_index	char(4)	transit_index
lcr_exp_upl_date	char(19)	lcr_exp_upl_date
legk_support	char(3)	legk_support, coded
dbsync_lock	char(3)	dbsync_lock, coded
cmi_max_subsc	char(5)	
pbx_upl_hour_adp	char(2)	
pbx_upl_min_adp	char(2)	
lcr_upl_hour_adp	char(2)	
lcr_upl_min_adp	char(2)	
pbx_upl_hour_mgr	char(2)	
pbx_upl_min_mgr	char(2)	
lcr_upl_hour_mgr	char(2)	
lcr_upl_min_mgr	char(2)	
tenant_group	char(20)	Note: **Manager only**
cmi_cos2	char(4)	
cmi_lcos2	char(4)	
cmi_lcosd	char(4)	

AM_VNR_SWITCHDATA		Switch data with VNR extensions
:EXPORT:SELECT		
cmi_lcosd2	char(4)	
wabegrpnum	char(2)	
vbzgrpnum	char(2)	
nunp_active	char(3)	coded
spe_support	char(3)	swu, coded
sec_tdm_subsr	char(3)	swu, coded
sec_tdm_trunk	char(3)	swu, coded
sec_ip_subsr	char(3)	swu, coded
sec_ip_trunk	char(3)	swu, coded
dns_ip_address	char(15)	swu
dls_ip_address	char(15)	swu
dls_port	char(5)	swu
key_len	char(3)	swu, coded
mikey_opt	char(3)	swu, coded
dls_accept_pass	char(3)	swu, coded
vnrad_active	char(3)	coded. Note: **Manager only**
vnr_switch_name	char(4)	Note: **Manager only**
seclvdsp	char(3)	swu, coded
upl_time_err	char(3)	coded. Note: **Manager only**
display_mode	char(3)	swu, coded
domain	char(8)	domain
virtual_node_id	char(10)	virtual_node_id, swu, key
partner	char(3)	Partner, swu, coded
default_entry	char(3)	default_entry, swu, coded
isdn_cc	char(4)	isdn_cc, swu
isdn_ac	char(6)	isdn_ac, swu
isdn_lc	char(10)	isdn_lc, swu
isdn_ul	char(3)	isdn_ul, swu, coded
isdn_skip_digits	char(2)	isdn_skip_digits, swu
pnp_l2	char(4)	pnp_l2, swu

AM_VNR_SWITCHDATA		Switch data with VNR extensions
:EXPORT:SELECT		
pnp_l1	char(6)	pnp_l1, swu
pnp_l0	char(10)	pnp_l0, swu
pnp_ul	char(3)	pnp_ul, swu, coded
pnp_skip_digits	char(2)	pnp_skip_digits, swu
virtual_node_num	char(6)	virtual_node_num, swu
skip_digits	char(1)	skip_digits, swu
cac_prefix	char(3)	cac_prefix, swu
sync_by	char(4)	
sync_time	int	
default_wabe	char(2)	swu
wabe0	char(3)	swu, coded
wabe1	char(3)	swu, coded
wabe2	char(3)	swu, coded
wabe3	char(3)	swu, coded
wabe4	char(3)	swu, coded
wabe5	char(3)	swu, coded
wabe6	char(3)	swu, coded
wabe7	char(3)	swu, coded
wabe8	char(3)	swu, coded
wabe9	char(3)	swu, coded
wabe10	char(3)	swu, coded
wabe11	char(3)	swu, coded
wabe12	char(3)	swu, coded
wabe13	char(3)	swu, coded
wabe14	char(3)	swu, coded
wabe15	char(3)	swu, coded

10.2.5.4 AMW_COMPORG

Table 231: AMW_COMPORG

AMW_COMPORG		Company/Organization data for AM-Win
:EXPORT:SELECT		
identifier	int	
country	char(2)	
company	char(30)	
address1	char(30)	
address2	char(30)	
org1	char(32)	
org2	char(32)	
org3	char(32)	
org4	char(32)	
org5	char(32)	
org6	char(32)	

10.2.5.5 AMW_PERSDATA

Table 232: AMW_PERSDATA

AMW_PERSDATA		Personal (subscriber) data for AM-Win
:EXPORT:SELECT:SELECT		
identifier	int	
country	char(2)	
company	char(30)	
org1	char(32)	
org2	char(32)	
org3	char(32)	
org4	char(32)	
org5	char(32)	
org6	char(32)	
name	char(25)	
sortname	char(30)	

AMW_PERSDATA		Personal (subscriber) data for AM-Win
:EXPORT:SELECT:SELECT		
christianname	char(15)	
title	char(15)	
salutation	char(15)	
location	char(5)	
building	char(5)	
longname	char(5)	
extension	char(12)	
switch	char(4)	
position	int	
tienum	char(6)	
ext_areacode	char(6)	
ext_netcode	char(16)	
ext_ext	char(6)	
faxnumber	char(6)	
owner_group	char(30)	
direct_trunk_acc	char(1)	
text1	char(24)	
text2	char(24)	
text3	char(24)	

10.2.5.6 AMW_PINDATA

Table 233: AMW_PINDATA

AMW_PINDATA		PIN data export for AM-Win
:EXPORT:SELECT:SELECT		
identifier	int	
domain	char(8)	
extension	char(12)	
switch	char(4)	
pin_num	char(12)	

AMW_PINDATA		PIN data export for AM-Win
:EXPORT:SELECT:SELECT		
pin_type	char(3)	
position	int	

10.2.5.7 AMW_SWITCHDATA

Table 234: AMW_SWITCHDATA

AMW_SWITCHDATA		Switch data export for AM-Win
:EXPORT:SELECT		
identifier	int	unique key
switch	char(4)	Name of the Hicom Switch
typ	char(10)	Switch ID of the system
version	char(5)	Version of the switch
info	char(30)	Additional info text
country	char(2)	Abbreviation of country ID
areacode	char(16)	Area code ("Ortskennzahl") of switch
netcode	char(16)	Net code ("Amtsnummer") of switch
idcard_plen	char(2)	SWU-data: max IDCARD PIN-nr
manual_plen	char(2)	SWU-data: max manual PIN length
passwd_plen	char(2)	SWU-data: max password length
amo_lang	char(3)	AMO language that will be used
nodenum	int	unique node number in Hicom V3.3 network
destnum	char(4)	destination number of switch
domain	char(8)	name of the domain network
vms_server	char(3)	server on which vms is running
vms_switch	char(4)	vms may run on another switch

10.3 Coded Values

10.3.1 General Coded Values (0-10)

0	No	nein
1	Yes	ja
2	Free	frei
3	Used	belegt
4	Cyclic	zyklisch
5	Linear	linear
6	German	deutsch
7	English	englisch
8	Card Subscriber	Ausweisteilnehmer
9	NON HICOM	NICHT-HICOM
10	not known	unknown

10.3.2 Call Log

11	Tries	Versuche

10.3.3 Signal

12	LED	LED

10.3.4 Displayanzeige

13	not extern	nicht extern
14	External Alternative Number	extern Ersatznummer
30	none	none
39	dds	dds
40	dss	dss

10.3.5 PIN Types (41-43)

41	Manual PIN	PIN manuell
42	ID Card PIN	Ausweis-PIN

43	Code for COS Switchover	BERUM-Code

10.3.6 General Coded Values (44-58)

44	small	small
45	large	large
46	noops	noops
*47	slma (duplicate of 300)	slma (duplicate of 300)
48	slma/t1	slma/t1
49	backup	backup
50	restore	restore
51	export	export
52	even	even
53	odd	odd
54	7	7
55	8	8
56	logged off	abgemeldet
57	Card subscriber signed on	Ausweistln. angemeldet
58	Cordless E STATION	Cordless E Teilnehmer

10.3.7 Error states (for action protocol)

60	Done without error	ohne Fehler beendet
61	Done with error	mit Fehler beendet
62	No response	keine Antwort
63	DMS Error	DMS - Fehler
64	Running	läuft
65	No Error	fehlerfrei
66	Error	Fehler
67	Done	beendet

10.3.8 Time Values

68	Daily	täglich
69	Monday	Montag
70	Tuesday	Dienstag
71	Wednesday	Mittwoch
72	Thursday	Donnerstag
73	Friday	Freitag
74	Saturday	Samstag
75	Sunday	Sonntag

10.3.9 Bus Types

76	mbus	mbus
77	cbus	cbus
78	fbus	fbus

10.3.10 General Coded Values (79 - 96)

79	opti	opti
*80	pnt (duplicate of 257)	pnt (duplicate of 257)
81	Limited	begrenzt
82	Unlimited	unbegrenzt
83	Unknown	unbekannt
84	Converted	umgewertet
85	Multiple seizure	mehrfach belegt
86	Not configured	nicht eingerichtet
87	Incorrect DESTNO	falsche ZLNR
88	Incorrect DAR	falscher KZP
89	Technical Station No.	technische Rufnummer
*90	IDCARDPIN (duplicate of 41)	IDCARDPIN (duplicate of 41)
*91	MANUALPIN (duplicate of 42)	MANUALPIN (duplicate of 42)

*92	BERUM (duplicate of 43)	BERUM (duplicate of 43)
93	swu	swu
94	vm	vm
95	multiple seizure/switched	mehrfach belegt/umgeschaltet
96	undefined	nicht definiert

10.3.11 Switches 100-149

100	HUNT Master	SSA Master
101	HTG	SSA
102	STN	TLN
103	ATNDDID	VPL
104	CO	AMT
105	TIE	QUER
106	TIEATND	QUVPL
107	SCONSCO	RFAMTUA
108	SCONSI	RFHAUSUA
109	SAT	VAMTUA

10.3.12 typ,rufnum

110	CTDL	STAFFELW

10.3.13 Switches 150-179

150	Non-Hicom	No Hicom
151	Hicom300	Hicom300
152	Hicom200	Hicom200
153	Hicom100	Hicom100
154	C80	C80
155	Hicom300R	Hicom300R

10.3.14 Port codec_type

170	G711 always to be used	Stets G711 benutzen
171	G711 preferable to be used	Möglichst G711 benutzen
172	G723 always to be used	Stets F723 benutzen
173	G723 preferable to be used	Möglichst F723 benutzen
174	G729 always to be used	Stets F729 benutzen
175	G729 preferable to be used	Möglichst F729 benutzen

10.3.15 Protocol types

180	1TR6	1TR6
181	Q931	Q931
182	DKZN1	DKZN1
183	DKZN2	DKZN2
184	CORNET	CORNET
185	SB1TR61	SB1TR61
186	SB1TR62	SB1TR62
187	SBQ9311	SBQ9311
188	SBQ9312	SBQ9312
189	SBDKZ1	SBDKZ1
190	SBDKZ2	SBDKZ2
191	SBEDSR1	SBEDSR1
192	SBEDSR2	SBEDSR2
193	SBEDSR3	SBEDSR3
194	SBEDSR4	SBEDSR4
195	SBDSS1	SBDSS1
196	NI-BRI	NI-BRI

10.3.16 *Devices 200-249

200	ANADP	ANAIW
201	ANADTMF	ANAMF
* 202	DIGI211 (duplicate of 551)	DIGI211 (duplicate of 551)
* 203	DIGI260 (duplicate of 550)	DIGI260 (duplicate of 550)
204	SET400	SET400
205	SET500	SET500
206	SET600	SET600
207	SET700	SET700
208	T3510	T3510
209	PCCARD	PCCARD
210	CLC	CLC
211	ANATE	ANATE
212	FAX	FAX
213	VTX	BTX
214	VTXA	BTXA
215	TTX	TTX
216	DTEA	DEEA
217	DTED	DEED
218	CTE3	CTE3
219	DTE	DEE
220	DCI500	DCI500
221	DCI700	DCI700
222	VTXD	BTXD
223	DTE/DCI700	DEE/DCI700
224	DTE/DCI500/DCI700	DEE/DCI500/DCI700
225	S0PP	S0PP
226	S2PP	S2PP
227	SET500/700	SET500/700
228	DIGI260/T3510	DIGI260/T3510

229	ANAVAR	ANAVAR
230	OPTI_E	OPTI_E
231	OPTI_phone_adaptor	OPTI_E_phone_adaptor
232	OPTI_ISDN_adaptor	OPTI_E_ISDN_adaptor
233	OPTI_E_data_adaptor	OPTI_E_data_adaptor
234	OPTI_E_control_adaptor	OPTI_E_control_adaptor
235	OPTI_ta/b_adaptor	OPTI_E_ta/b_adaptor
236	HTBMNGR	HTBMNGR
237	HTBPHONE	HTBPHONE
238	HTBBEXT	HTBVLV
239	HTBLINE	HTBLINE
240	ANATECM	ANATEGEB
241	TCLOG	TCLOG
242	TCVLV	TCVLV
243	MSN	MSN
244	API	API
245	AUXDIGI	AUXDIGI
246	DCI400	DCI400
247	NTVLV	NTVLV
248	REMOPTI	REMOPTI
249	ANADEV	ANAEG

10.3.17 Feature 250-269

250	OFF	OHNE
251	ON3	MIT3
252	IDLE3	RUHE3
253	ON1	MIT1
254	IDLE1	RUHE1
256	DIRECT	DIRECT
257	PNT	PNT
258	INCOMING	INCOMING

259	OUTGOING	OUTGOING
260	EXTERNAL	EXTERNAL
261	PERIOD3	PERIOD3
262	PERIOD1	PERIOD1
263	LOG	LOG
264	ATMPSW	ATMPSW
265	ATMTFA	ATMTFA
266	OPTI-IP	OPTI-IP
267	IP	IP

10.3.18 Feature 270-299

270	Exec.1	Chef1
271	Exec.2	Chef2
272	Exec.3	Chef3
273	Exec.4	Chef4
274	Secr.1	Sekretär1
275	Secr.2	Sekretär2
276	Subst.	Vertretung
277	Subst.2	Vertretung2
278	Exec.	Chef
279	Secr.	Sekretär
280	Executive Line	Chefleitung

10.3.19 Boards 300-349

300	SLMA	SLMA
301	SLMB	SLMB
302	SLMD	SLMD
303	TMX21	TMX21
304	SLMU	SLMU
305	SLMS	SLMS
306	STMD	STMD

307	DIU-S2	DIUS2
308	SLMQ	SLMQ
309	DIU24	DIU24
310	DIU30	DIU30

10.3.20 ROLM boards 310-330

311	SLMR	SLMR
312	ATI	ATI
313	OPS	OPS
314	T1	T1
315	RLI	RLI
316	TMD24	TMD24
317	SLMAVAR	SLMAVAR
318	SLMQ_EXT	SLMQ_EXT
319	SLMO	SLMO
320	SLMA1	SLMA1
321	SLMU16	SLMU16
330	KLMU	KLMU

10.3.21 CMI board

331	SLCE8	SLCE8
332	SLC16	SLC16
333	SLC16-100	SLC16-100
334	SLC16-200	SLC16-200
335	SLC16-300	SLC16-300
336	SLC24	SLC24
337	SLCSM	SLCSM

10.3.22 Boards for V3.6

340	SLMY	SLMY

341	WAML2	WAML2
342	STMA	STMA
343	SLMAR	SLMAR
344	SLMO_ROUT_TEMP	SLMO_ROUT_TEMP
345	SLMO_REM_TEMP	SLMO_REM_TEMP
346	SLMO_ROUT_PERM	SLMO_ROUT_PERM
347	SLMO_REM_PERM	SLMO_REM_PERM
348	STMA_PSW	STMA_PSW

10.3.23 Boards for DMS3.6

349	SLMOP	SLMOP
350	STMD2	STMD2
351	SLMS2	SLMS2
352	STHCO	STHCO
353	STHCD	STHCD

10.3.24 Boards for Unity

355	DIU-N2	DIU-N2
356	DIU-N4	DIU-N4
357	SLMQ3	SLMQ3
358	SLMA2	SLMA2
359	TMDNH	TMDNH
360	STMI-HFA	STMI-HFA
361	STMI-IGW	STMI-IGW
362	APPS	APPS
363	BS22	BS22
364	CDG	CDG
365	CDG31-DE	CDG31-DE
366	CDG31-FU	CDG31-FU
367	CDG31-RG	CDG31-RG
368	CGM&NEW	CGM&NEW

369	CSG	CSG
370	DIUC64	DIUC64
371	DPC5	DPC5
372	DSCX	DSCX
373	LTUCX	LTUCX
374	MTSCG+CB	MTSCG+CB
375	NCUI	NCUI
376	PBCDG-DE	PBCDG-DE
377	PBCDG-FU	PBCDG-FU
378	PBCDG-RG	PBCDG-RG
379	PBPNE-DE	PBPNE-DE
380	PBPNE-RG	PBPNE-RG
381	PNEDIUS2	PNEDIUS2
382	RG-USA	RG-USA
383	RGKOR	RGKOR
384	SIUX2	SIUX2
385	SLMA3	SLMA3
386	SLMAB	SLMAB
387	SLMO16	SLMO16
388	SLOP2	SLOP2
389	SO-BG	SO-BG
390	STHC	STHC
391	STHC2	STHC2
392	TM2LP	TM2LP
393	TM3WI	TM3WI
394	TM3WO	TM3WO
395	TMACH	TMACH
396	TMAU	TMAU
397	TMBD	TMBD
398	TMBLN	TMBLN
399	TMCOW	TMCOW
400	TMEM	TMEM

401	TMEW2	TMEW2	
402	TMLBL	TMLBL	
403	TMLR	TMLR	
404	TMLRB	TMLRB	
405	TMLSL	TMLSL	
406	TMSFP	TMSFP	
407	VCM-B15	VCM-B15	
408	VCM-B7	VCM-B7	
409	PNE	PNE	* m30 (ASU01)
410	STMI	STMI	* d31 (ASU01)
411	STMI2-HFA	STMI2-HFA	
412	DIUN2U	DIUN2U	

*			

418	ST	ST	
424	VC__	VC	
428	PUS	AUNS	
429	MSGR	BOTE	
430	NV	NV	
431	RNGXFER	RU	
432	REMIND	TE	
433	RLS	TR	
434	DOOR	TUER	
435	REP	VTR	
436	ADDON	ZZU	

***Boards continuation (440-460)**

* 440	HUBC	HUBC	* t74 (DMS60)
* 441	LTUCE	LTUCE	* u51 (DMS60)
* 442	MTSCG	MTSCG	* u87 (DMS60)
* 443	PSIO	PSIO	* t71 (DMS60)

* 444	QDCL	QDCL	* u98 (DMS60)
* 445	RG	RG	* u43 (DMS60)
* 446	SCC	SCC	* u52 (DMS60)
* 447	SICOE	SICOE	* u99 (DMS60)
* 448	SIUX	SIUX	* u41 (DMS60)
* 449	SLMA24	SLMA24	* v11 (DMS60)
* 450	SLMB4	SLMB4	* u22 (DMS60)
* 451	SLMO24	SLMO24	* u24 (DMS60)
* 452	TMC16	TMC16	* u32 (DMS60)
* 453	TMC16P	TMC16P	* v07 (DMS60)
* 454	TMDID	TMDID	* u33 (DMS60)
* 455	TMEMUS	TMEMUS	* u36 (DMS60)
* 456	TMEMW	TMEMW	* u35 (DMS60)
* 457	TMGSR	TMGSR	* u37 (DMS60)
* 458	TMOM2	TMOM2	* v15 (DMS60)
* ----------			
461	MUTE	MUTE	

***Boards continuation (462-489)**

* ----------		
497	HOT	HOT
503	MB__	MB
504	LNR	WAWIL

10.3.25 headset,devconst

510	No Indication	keine Indikation
511	With Indication	mit Indikation

10.3.26 recorder,devconst

512	Indication	Indikation

513	Normal	Normal

10.3.27 pmm parameter definitions

530	American English	American English
531	Canadian English	Canadian English
532	Japanese	Japanese
533	UK English	UK English
534	german	German
535	American English TDD	American English TDD

* 536 Unknown (duplicate of 10)

10.3.28 Server 537-549

537 T1 (duplicate of 314)

538	T2	T2
539	T3	T3
540	V1	V1
541	V2	V2
542	V3	V3
543	ADS	ADS

546	OPTI_E_T3	OPTI_E_T3
547	OPTI_E_T8	OPTI_E_T8
548	OPTI_E_T12	OPTI_E_T12

10.3.29 device types

550	DIGI260	DIGI260
551	DIGI211	DIGI211
552	SET4ST	SET4ST
553	SET4C1	SET4C1
554	SET4C2	SET4C2
555	SET5B	SET5B

556	SET5B1	SET5B1
557	KEY300	KEY300

10.3.30 Optiset conference

559	OPTI_E_CONFERENCE	OPTI_E_CONFERENCE

10.3.31 exact device types

560	DIGITE211	DIGITE211
561	DIGITE260	DIGITE260

* 562 T3510 (duplicate of 208)

563	SET421T16D	SET421T16D
564	SET421T24FD	SET421T24FD
565	SET451T8	SET451T8
566	SET451T8R	SET451T8R
567	SET451T24D	SET451T24D
568	SET451T25FD	SET451T25FD
569	SET551T17	SET551T17
570	SET551T25FD	SET551T25FD
571	SET751T25FD	SET751T25FD
572	SET551_751T25FD	SET551_751T25FD
573	OPTI_E_ENTRY	OPTI_E_ENTRY
574	OPTI_E_SIGNATURE	OPTI_E_SIGNATURE
575	OPTI_E_BASIC	OPTI_E_BASIC
576	OPTI_E_STANDARD	OPTI_E_STANDARD
577	OPTI_E_ADVANCE	OPTI_E_ADVANCE
578	OPTI_E_ADVANCE+	OPTI_E_ADVANCE+
579	OPTI_E_SELECT	OPTI_E_SELECT
580	OPTI_E_MEMORY	OPTI_E_MEMORY

581	CPP	CPP
582	FBUS	FBUS

583	CBUSEXP	CBUSEXP
584	MBUSEXP	MBUSEXP
585	FPP	FPP
586	FPPEXP	FPPEXP
587	SVC	SVC
588	NT	NT
589	HTB	HTB
590	TC	TC

10.3.32 Addon type

591	1_OPTI_KEY_MOD	1_OPTI_KEY_MOD
592	2_OPTI_KEY_MOD	2_OPTI_KEY_MOD
593	3_OPTI_KEY_MOD	3_OPTI_KEY_MOD
594	4_OPTI_KEY_MOD	4_OPTI_KEY_MOD

595	SETT29	SETT29
596	SET410N	SET410N
597	SET410T	SET410T
598	1 OPTIP KEY MOD SLK	1 OPTIP KEY MOD SLK
599	2 OPTIP KEY MOD SLK	2 OPTIP KEY MOD SLK

10.3.33 Object_type for action_protocol 600-630

600	System	Anlage
601	Station	Teilnehmeranschluß
602	Device Combination	Gerätekombination
604	Call Pickup	Anrufübernahme
605	PU Remote Links	AUN Netzverbindungen
606	Hunting Station	Sammelanschlußteilnehmer
607	Executive / Secretary Group	Chef-Sekretär Gruppe
609	Company	Firma
611	Location	Standort

613	Subscriber	Teilnehmer
614	Extension	Rufnummer
615	COS Changeover Group	Berechtigungsumschaltegruppe
616	Class of Service	Berechtigungsklasse
617	Speed Dial List	Kurzrufnummernliste
618	Sample Station	Musterteilnehmer
619	VMS Class of Service	VMS-Berechtigungsklasse
620	Automatic Download	Automatischer Download
621	Manual Download	Manueller Download
622	Standard Key Layout	Standard-Tastenbelegung
623	Individual Key Layout	Individuelle Tastenbelegung
624	Executive / Secretary Group Member	Chef / Sekretär Gruppenmitglied
625	External COS (COEX)	Externe COS (COEX)
626	Names / Departments	Namen / Abteilungen
627	Error Log	Fehlerprotokoll
628	Session Log	Sitzungsprotokoll
629	System Log	Anlagenprotokoll
630	Action Control	Aktionskontrolle

10.3.34 Operations for action_protocol

631	Add	Einrichten
632	Delete	Löschen
633	Modify	Ã„ndern
634	Information	Information
635	Single-phase Move	Umzug einphasig
636	Two-phase Move 1st Phase	Umzug zweiphasig 1.Phase
637	Two-phase Move 2nd Phase	Umzug zweiphasig 2.Phase
638	Two-phase Move Undo	Umzug zweiphasig Abbruch
639	Adapt	Anpassen

640	Save	Sichern
641	Upload All	Upload-All
642	Upload Delta	Upload-Delta
643	Remove Lock	Rücknahme Sperre
644	VMS Upload All	VMS-Upload-All
645	VMS Upload Delta	VMS-Upload-Delta
646	Include	Einschliessen
647	Data transfer	Datentransfer
648	Verify	Prüfen
649	Add Name/PINs	Einrichten Name/PINs
650	Delete Name/PINs	Löschen Name/PINs
651	Modify Name/PINs	Ã„ndern Name/PINs
652	Add Member	Mitglied hinzufügen
653	Delete Member	Mitglied herausnehmen
654	Modify Member	Mitglied ändern
655	Add Building	Gebäude hinzufügen
656	Delete Building	Gebäude entfernen

658	Add Organization Unit	Organisationseinheit hinzufügen
659	Remove Organization Unit	Organisationseinheit entfernen
661	Add VMS Mailbox	Einrichten VMS Mailbox
662	Delete VMS Mailbox	Löschen VMS Mailbox
663	Modify VMS Mailbox	Ã„ndern VMS Mailbox
670	LCR Upload-All	LCR-Upload-All
671	LCR Upload-Delta	LCR-Upload-Delta

10.3.35 Object_type for action_protocol for LCR (678-690)

678	Trunk Group	Bündel
679	LCR Authorization	LCR-Berechtigungen
680	LCR Class of Service - Data	LCR-Berechtigungsklasse Daten

681	LCR Class of Service - Voice	LCR-Berechtigungsklasse Sprache
682	LCR Route Element	LCR-Richtungselement
683	LCR Numbering Plan	LCR-Rufnummernplan
684	LCR Numbering Plan for DPLN Group	LCR-Rufnummernplan für WABE-Gruppe
685	LCR Outdial Rule	LCR-Wahlregel
686	LCR Outdial Rule Element	Element einer LCR-Wahlregel
687	LCR Timetable	LCR-Zeitplan
688	Zone Information	Zoneninformation
689	LCR Route	LCR-Richtung
690	LCR Route Trunk Group	LCR-Richtungsbündel
691	Cordless Global	Cordless-Global
692	Cordless Global Sub	Cordless-Global-Sub
693	Cordless System	Cordless-System
694	Cordless System Sub	Cordless-System-Sub
695	Cordless Base Station	Cordless-Basisstation
696	Networkwide Team	Netzweites Team

10.3.36 Rufnummernplan

698	Type1	Typ1
699	Type2	Typ2

10.3.37 DIMSU's

700	Port	Teiln. Anschl.
701	Hunting	Sammelanschl.
702	Display Names	Display Namen
703	Digital Devices	digitale Geräte
704	SET260 Devices	SET260 Geräte
705	SET400 Devices	SET400 Geräte
706	SET500/700 Devices	SET500/700 Geräte

707	Classes of Service	Berecht. Klassen
708	PU Groups	AUN Gruppen
709	Call Queues (HT)	Anrufordner (SA)
710	PINs	PINs
711	Spd Dial Ind.	Kurzw. Indiv.
712	Mailboxes	Postfächer
713	Mailb. w/ PFTyp1	Postf. m. PFTyp1
714	Mailb. w/ PFTyp2	Postf. m. PFTyp2
715	Mailb. w/ PFTyp3	Postf. m. PFTyp3
716	Mailb. w/ PFTyp4	Postf. m. PFTyp4
717	Mailb. w/ ASTyp1	Postf. m. ASTyp1
718	Mailb. w/ ASTyp2	Postf. m. ASTyp2
719	Mailb. w/ ASTyp3	Postf. m. ASTyp3
720	Mailb. w/ ASTyp4	Postf. m. ASTyp4
721	VMS COS	VMS Berecht. Kl.
722	Brdcast. Dist. List	Rundspruchverteiler
723	Spd. Dial Lists	Kurzwahl Listen
724	Spd. Dial Dests.	Kurzwahl Ziele
725	DSS Keys	Direktruftasten
726	SO Bus Config.	SO Bus Konfig.
727	Add-on Devices	Beistellgeräte
728	Exec/Secr groups	CHESE Gruppen
729	Members of Exec/Secr	Mitgl. in CHESE
730	Extended Connections	Verlängerungsverbindungen

10.3.38 additional dimsu

731	Call Forward - DSS1	Anrufumlenkung DSS1
732	ATM Port	ATM Port
733	Optisets with Call Logging	Optisets mit Rufjournal
734	Application with WS Protocol	Applikation mit WS-Protokoll
735	Key System	Key System

736	Info	Info
737	Remote PU	Remote AUN

10.3.39 Unity's dimsu

738	Attendant consoles	Vermittlungsplätze
739	di branch-offs for s2 trunks	di-Verzweigungen für s2-Leitungen
740	Trunks (analog, digital) + b-channel groups for s2	Externe Leitungen (analog, digital) + B-Kanalgruppe für s2
741	Dial plan blocks for access codes	Wählplan-Blöcke für Kennzahlen
742	Dial plan blocks for dial plan group branches	Wählplan-Blöcke für Wählplangruppenverzweigung
743	Routes	Richtungen
744	Master groups for device search	Master-Gruppen für Gerätesuche
745	Subgroups for device search	Untergruppen für Gerätesuche
746	Attendant console groups	VPL-Gruppen
747	Organizational units	Organisationseinheiten
748	Dynamically assignable pool elements for callback	Dynamisch zuweisbare Rückruf-Poolelemente
749	Timed reminder entries	Terminruf-Einträge
750	Dynamically assignable cp connection buffers	Dynamisch zuweisbare VT-Verbindungspuffer
751	Entries in the list for saved connections	Einträge in der Liste der gespeicherten Verbindungen
752	Connections with charge data recording	Verbindungen mit Gebührenerfassung
753	Dynamic pin buffers for networkwide identification	Dynamisch PIN-Puffer für netzweite Identifizierung
754	User profiles for closed user groups	Benutzerprofile für geschlossene Benutzergruppen

755	Blocks of 5 dss partners for storing dss partner inf.	Blöcke von 5 DSS-Partner zur Speicherung der DSS-Partner-Information	
756	Dyn. call-queuing pool elements for attendant groups	Dynamische Anrufordner-Poolelemente für VPL-Gruppen	
757	Night stations for special night answer position	Teilnehmeranschluß für spezielle Nachtstelle	
758	MFC-variants for analog trunks and tie-lines	MFC-Variante für analoge Leitungen und Querverbindungen	
759	Display modifications between network nodes	Ã„nderungen zwischen Netzwerkknoten anzeigen	
760	Datapool elements reserved for dimsu param. dspmod	Datenpool-Elemente für DIMSU-Parameter 'dspmod'	reserved
761	Number of dialplan entries for LCR in pool 1	Anzahl Wahlberechtigungseinträge für LCR im Pool 1	
762	Number of entries in the LCR dial plan cache for pool 1	Anzahl Einträge im LCR-Wahlberechtigungs-Cache für Pool 1	
763	Number of outdial rules	LCR Wahlregeln	
764	Dyn. announcement buffers for synchronous announcement	Dynamische Ansagepuffer für synchrone Ansagen	
765	Route elements for LCR	Richtungselemente für LCR	
766	Entries in the format table for subscriber numbers	Einträge in der Formattabelle für Teilnehmernummern	
767	Compressed b-channels for voice connections	Komprimierte B-Kanäle für Sprachverbindungen	
768	Sort elements for nx64 kbits/s feature	Sortier-Elemente für Funktion 'nx64 kbits/s'	
769	100-blocks with dialing numbers for all stations	100-Blöcke mit Rufnummern für alle Teilnehmeranschlüsse	
770	Restricted connections from originating nodes	Eingeschränkte Verbindungen von den Ursprungknoten	

771	Screening data records for incoming e-dss1 conn.	Auswahl-Datensätze für kommende E-DSS1-Verbindungen
772	Screening patterns for e-dss1 connections	Auswahlmuster für E-DSS1-Verbindungen
773	DP applications (acl-h3 and acl-c)	DP-Applikationen (ACL-H3 und ACL-C)
774	Ext. computer-controlled call setups (acl-h3, acl-c)	Ext. computer-gesteuerter Verbindungsaufbau (ACL_H3, ACL-C)
775	ACD monitor points for acl-h3 applications	ACD-Monitor-Punkte für ACL-H3-Applikationen
776	Parameter sets for dci	DCI-Parametersätze
777	Parameter sets for modem pool	Modem-Pool-Parametersätze
778	Call-related data for the workstation protocol	Anrufbezogene Daten für 'Workstation-Protokoll'
779	Datapool elements reserved for dimsu parameter info	Datenpool-Elemente für DIMSU-Parameter 'info' reserviert
780	Peripheral hardware modules	Peripherie-Hardwaremodule
781	Common connection buffers for multiple terminal conn.	Allg. Verbindungspuffer für mehrfache Geräteverbindungen
782	WAML boards for routing	WAML-Baugruppen für Routing
783	WAML boards for telecommuting	WAML-Baugruppen für Telearbeit
784	STMA boards	STMA-BG
785	TFA groups of all stma boards with operating mode tfa	TFA-Gruppen aller STMA-Baugruppen mit Betriebsmodus 'tfa'
786	ACD agents	ACD-Agenten
787	ACD groups	ACD-Gruppen
788	ACD agents logged on at once	ACD-Agenten sofort angemeldet
789	Primary acd agent positions	Primäre ACD-Agentenpositionen

790	Dyn. assignable acd call-queuing pool elements	dyn. zuweisbare ACD-Anrufordner-Pool-Elemente
791	ACD route-control groups	ACD-Route-Control-Gruppe
792	ACD routing step tables for flexrouting	ACD-Routing-Schritttabellen für Flexrouting
793	Auxiliary acd routing step tables	Zusatz-ACD-Routing-Schritttabellen
794	ACD shifts	ACD-Schichten
795	Monitored id sets (e.g. acd agents - only acd-g)	Ãberwachte Id-Satz (z.B. ACD-Agenten - nurACD-G)
796	Calendar routing entries	Kalender-Routing-Einträge
797	Entries in the gdtr table	Einträge in der GDTR-Tabelle
798	Directory number information table indices	Indices der Rufnummer-Informationstabellen
799	Datapool elements reserved for dimsu parameter dni	Datenpool-Elemente reserviert für DIMSU-Parameter 'dni'
800	Additional connection buffers for acd-g	Zusätzliche Verbindungspuffers für ACD-G
801	Permanently configured device table entries	Tabelleneinträge für fest konfigurierte Geräte
802	Number of Optiset/ Optipoint busses, S0 busses and S0 trunks	Anzahl der Optiset/ Optipoint-Busse, S0-Busse und S0-Leitungen
803	B-channels for trunks + analog trunks/tie-lines	B-Kanäle für externe Leitungen + analoge externe Leitungen/ Querverbindungen
804	Dynamically assignable device table entries	Tabelleneinträge für dynamisch zuweisbare Geräte
805	Device table entries for d-channel conn. of s0/s2	Gerät-Tabelleneinträge für D-Kanal-Verbindung der s0/s2
806	Dynamically assignable cp data buffers	Dynamisch zuweisbare VT-Datenpuffers

807	Call reference tables in cp	Call-Reference-Tabellen in VT
808	Stations which are member of a hunt group	Sammelanschluss-Teilnehmer
809	Elements for call forwarding history	Elemente für Anrufumleitung-Historie
810	Location identification numbers for emergency calls	LINs für Notrufe
811	Blocked digit strings for lcr route elements	Blockierte Ziffernfolge für LCR-Richtungselemente
812	DBMA index for static CC-DB (CCIDXS)	DBMA-Index für statische CC-DB (CCIDXS)
813	DBMA index for dynamic CC-DB (CCIDXD)	DBMA-Index für dynamische CC-DB (CCIDXD)
814	DBMA index for GP-DB (GPIDX)	DBMA-Index für GP-DB (GPIDX)
815	Lines	Leitungen
816	Ports	Anschlüsse
817	Number of STMI boards for IP device connections	Anzahl der STMI-Baugruppen für IP-Gerätverbindungen
818	Number number of stmi boards for access point connection	Anzahl der STMI-Baugruppen für Access-Point-Verbindung
819	Number of access points	Anzahl Access Points
820	Profiles for source depending routing	Profile für ursprungsabhängiges Routing
821	No. of device table entries (DTI) for trunks and terminals	Anzahl der Gerätespeicher (DTI) für Sätze und Endgeräte
822	Dev. table entries for B/D-chan. conn. of S0/S2 trunks	Anz. Gerätespeicher für B/D-Kanal-verb. von S0/S2-Leitungen
823	No. of subscriber names for ext. call forwarding destinations	Anzahl von Teilnehmernamen für externe Anrufumleitungsziele

824	Number of lines to partner nodes	Anzahl der Verbindungen zu Partnerknoten	
825	Number of STMI2 boards for IP device connections	Anzahl der STMI2-Baugruppen für IP-Geräteverbindungen	
826	Number of SelfLabellingKeys SLK with individual text	Anzahl der SLK mit individuellem Text	
828	Number of optiPoint devices requiring shadow call log	Anzahl der optiPoint-Geräte mit Zwiechenspeicherung	
830	Individual alternate routing numbers	Individuelle alternative Rufnummern	
831	Number of LCR Class of Services	Anzahl LCR-COS	
832	Hotline Index	Hotline Index	
833	DIMSU_APE_AP CC	Anzahl der CC-APs für AP Emergency	
834	DIMSU APE EM GRP	Anzahl der AP Emergency Gruppen	
840	Call forwarding to network	Anrufumleitung extern	* dimsu for pre-unity systems (unity: code 731)

10.3.40 fixed call forwarding

841	CFU	CFU	
842	CFB	CFB	
843	CFNR	RWS	
844	CD	CD	
845	CFDEV	CFDEV	
846	CFDND	CFDND	
847	CFBNR	CFBNR	

10.3.41 key functions (850-899)

850	DND	ANT
851	KNOVR	ASAK
852	VCR	ASSCH
853	FWD	AUL
854	PU	AUN

* 855 PUS/AUNS (duplicate of 428)

856	MB	BRK

* 857 MSGR/BOTE (duplicate of 429)

858	CH	CH
859	CL	CL
860	VC	DA
861	DSS	DR
862	STO	DUE

* 863 FREE/FREI (duplicate of 2)

864	CONF	KF
865	LNAME	LNA
866	SPKR	LS
867	SPLT	MA

* 868 MUTE (duplicate of 461)

869	NAME	NA

* 870 NV (duplicate of 430)

871	PARK	PA
872	CC	PS
873	CONS	RF
874	CBK	RR

* 875 RNGXFER/RU (duplicate of 431)

876	HT	SA
877	HEADSET	SG

* 878 ST (duplicate of 418)
* 879 REMIND/TE (duplicate of 432)
* 880 RLS/TR (duplicate of 433)
* 881 DOOR/TUER (duplicate of 434)

882	TRNS	UEG
883	TIME	UHR

* 884 REP/VTR (duplicate of 435)

885	SNR	WW

* 886 ADDON/ZZU (duplicate of 436)

887	IUSE	ANZLEIT
888	LINE	LEITUNG
889	PRIM	PRIM
890	EXHOLD	EXHALTEN
891	HOLD	HALTEN
892	PRIVACY	PRIVAT
893	RCUTOFF	RUFAUS
894	SHIFT	UMSCHALT
895	MBLINE	BKLEITNG
896	RNGXFERL_DEST	RULEITNG_ZIEL
897	RNGXFERL_SRC	RULEITNG_URSPRUNG
898	RNGXFERL	
899	MONSLNTL	

900	Optisets	Optisets
901	Addon + ID Card Reader + Control Adaptor	Beistellg. + Ausweisl. + Control Adaptor

10.3.42 key functions (920-949)

920	ACDLOG	ACDLOG
921	ACDAV	ACDVB
922	ACDWORK	ACDARB
923	ACDNAV	ACDNVB

924	ACDPGS	ACDPGS	
925	ACDSGS	ACDSGS	
926	ACDPQS	ACDPQS	
927	ACDSQS	ACDSQS	
928	ACDSPVN	ACDSPVN	
929	ACDAGN	ACDAGN	
930	ACDEMMSG	ACDNOTN	
931	SCROL	SCROL	
932	ACK	ACK	
933	LANSEL	SPRWA	
934	MONTONE	MITHOERT	
935	AUTOM	AUTOM	
936	ACC	ACC	
937	DTB	DTB	
938	DTB_CL_INC	DTB_CL_INC	* German: DTB_RJ_ANK
939	DTB_CL_OUT	DTB_CL_OUT	* German: DTB_RJ_ABG
940	CALL_LOG	CALL_LOG	

10.3.43 Code values for PORT / net_config

960	Logical Subscriber	Logischer Teilnehmer
961	Remote Subscriber	Abgesetzter Teilnehmer
962	Extended Connection	Verlängerungsverbindung
963	HTB System Manager	HTB-System-Manager

10.3.44 pickup_sub, pararng and mastergrp

970	Master	Master	
* 971	Subscriber	Subscriber	* (duplicate of 613)

10.3.45 *Coded values for LCR-feature

buend, trunk_group_type

A00	special	spezial
A01	digital	digital
A02	analogue	analog

buend, traffic_type

A03	incoming	kommend
A04	outgoing	gehend
A05	both	beidseitig

buend, sat_count

A06	not any	keine
A07	one	eine
A08	two	zwei
A09	many	viele

lcos_voice, lcr_opt

A10	expensive	teuer

* A11 not any / keine (duplicate of A06)

lcr, zone

A12	empty	leer
A13	Local	Lokal
A14	Short Distance	Nahbereich
A15	internal01	intern01
A16	internal02	intern02
A17	internal03	intern03
A18	internal04	intern04
A19	internal05	intern05
A20	internal06	intern06
A21	internal07	intern07
A22	internal08	intern08
A23	internal09	intern09

A24	Long01	Fern01
A25	Long02	Fern02
A26	Long03	Fern03
A27	Long04	Fern04
A28	Long05	Fern05
A29	Long06	Fern06
A30	Long07	Fern07
A31	Long08	Fern08
A32	Long09	Fern09
A33	Long10	Fern10
A34	Long11	Fern11
A35	Long12	Fern12
A36	Long13	Fern13
A37	Long14	Fern14
A38	Long15	Fern15
A39	Long16	Fern16
A40	Long17	Fern17
A41	Long18	Fern18
A42	Long19	Fern19
A43	Long20	Fern20
A44	Long21	Fern21
A45	Long22	Fern22
A46	Long23	Fern23
A47	Long24	Fern24
A48	Long25	Fern25
A49	Long26	Fern26
A50	Long27	Fern27
A51	Long28	Fern28
A52	Long29	Fern29
A53	Long30	Fern30
A54	Long31	Fern31
A55	Long32	Fern32

A56	Long33	Fern33
A57	Long34	Fern34
A58	Long35	Fern35
A59	Long36	Fern36
A60	Long37	Fern37
A61	Long38	Fern38
A62	Long39	Fern39
A63	Long40	Fern40
A64	Long41	Fern41
A65	Long42	Fern42
A66	Long43	Fern43
A67	Long44	Fern44
A68	Long45	Fern45
A69	Long46	Fern46
A70	Long47	Fern47
A71	Long48	Fern48
A72	Long49	Fern49
A73	Long50	Fern50

*

A80	Tie01	Quer01
A81	Tie02	Quer02
A82	Tie03	Quer03
A83	Tie04	Quer04
A84	Tie05	Quer05
A85	Tie06	Quer06
A86	Tie07	Quer07
A87	Tie08	Quer08
A88	Tie09	Quer09

*

B01	Interntl01	Ausland01
B02	Interntl02	Ausland02

B03	Interntl03	Ausland03
B04	Interntl04	Ausland04
B05	Interntl05	Ausland05
B06	Interntl06	Ausland06
B07	Interntl07	Ausland07
B08	Interntl08	Ausland08
B09	Interntl09	Ausland09
B10	Interntl10	Ausland10
B11	Interntl11	Ausland11
B12	Interntl12	Ausland12
B13	Interntl13	Ausland13
B14	Interntl14	Ausland14
B15	Interntl15	Ausland15
B16	Interntl16	Ausland16
B17	Interntl17	Ausland17
B18	Interntl18	Ausland18
B19	Interntl19	Ausland19
B20	Interntl20	Ausland20
B21	Interntl21	Ausland21
B22	Interntl22	Ausland22
B23	Interntl23	Ausland23
B24	Interntl24	Ausland24
B25	Interntl25	Ausland25
B26	Interntl26	Ausland26
B27	Interntl27	Ausland27
B28	Interntl28	Ausland28
B29	Interntl29	Ausland29
B30	Interntl30	Ausland30
B31	Interntl31	Ausland31
B32	Interntl32	Ausland32
B33	Interntl33	Ausland33
B34	Interntl34	Ausland34

B35	Interntl35	Ausland35
B36	Interntl36	Ausland36
B37	Interntl37	Ausland37
B38	Interntl38	Ausland38
B39	Interntl39	Ausland39
B40	Interntl40	Ausland40
B41	Interntl41	Ausland41
B42	Interntl42	Ausland42
B43	Interntl43	Ausland43
B44	Interntl44	Ausland44
B45	Interntl45	Ausland45
B46	Interntl46	Ausland46
B47	Interntl47	Ausland47
B48	Interntl48	Ausland48
B49	Interntl49	Ausland49
B50	Interntl50	Ausland50
B51	Interntl51	Ausland51
B52	Interntl52	Ausland52
B53	Interntl53	Ausland53
B54	Interntl54	Ausland54
B55	Interntl55	Ausland55
B56	Interntl56	Ausland56
B57	Interntl57	Ausland57
B58	Interntl58	Ausland58
B59	Interntl59	Ausland59
B60	Interntl60	Ausland60
B61	Interntl61	Ausland61
B62	Interntl62	Ausland62
B63	Interntl63	Ausland63
B64	Interntl64	Ausland64
B65	Interntl65	Ausland65
B66	Interntl66	Ausland66

B67	Interntl67	Ausland67
B68	Interntl68	Ausland68
B69	Interntl69	Ausland69
B70	Interntl70	Ausland70
B71	Interntl71	Ausland71
B72	Interntl72	Ausland72
B73	Interntl73	Ausland73
B74	Interntl74	Ausland74
B75	Interntl75	Ausland75
B76	Interntl76	Ausland76
B77	Interntl77	Ausland77
B78	Interntl78	Ausland78
B79	Interntl79	Ausland79
B80	Interntl80	Ausland80
B81	Interntl81	Ausland81
B82	Interntl82	Ausland82
B83	Interntl83	Ausland83
B84	Interntl84	Ausland84
B85	Interntl85	Ausland85
B86	Interntl86	Ausland86
B87	Interntl87	Ausland87
B88	Interntl88	Ausland88
B89	Interntl89	Ausland89
B90	Interntl90	Ausland90
B91	Interntl91	Ausland91
B92	Interntl92	Ausland92
B93	Interntl93	Ausland93
B94	Interntl94	Ausland94
B95	Interntl95	Ausland95
B96	Interntl96	Ausland96
B97	Interntl97	Ausland97
B98	Interntl98	Ausland98

B99	Interntl99	Ausland99

lcr, lcr_attribute

C01	Not any	Keine
C02	TRANSGR	TRANSGR
C03	NATNUM	NATNUM
C04	INTERNUM	INTERNUM
C05	SUBSCNUM	TLNNUM
C06	NUMPLAN	NUMPLAN
C07	WCHREG	GEBAKT
C08	NOTRANS	TRANSOG
C09	NOCHDISP	EXPWARN
C10	INTCHARG	EXTINT
C11	EODSTCDR	GEBSTWE
C12	MODEM	MODEM

lcr, voice

* C13	Sprache	Sprache	(duplicate of ???)
* C14	Daten	Daten	(duplicate of ???)
* C15	Spr&Daten	Spr&Daten	(duplicate of ???)

lrtg, type

C16	LCR	LCR
C17	NWLCR	NWLCR
C18	All	Alle

lrtg, mfvums

C19	off	ohne
C20	fix	fix
C21	Suffix dial	Nachwahl

lcrrule_sub, rule_element

C22	OUTCODE	CODESEND
C23	OUTCOEX	OUTCOEX

C24	OUTCSN	CSNSEND
C25	ECHO	ECHOFELD
C26	ECHOALL	ECHOREST
C27	END	ENDE
C28	DTMF	MFVUMSCH
C29	PAUSE	PAUSE
C30	WAITANS	WTONWART
C31	OUTPULSE	ZIFFSEND
C32	SENDCOMP	SENDCOMP

*

C33	DTMF Text	MFV-Text
C34	Digit	Ziffer
C35	Digits	Ziffern

*

C36	SENDNOW	SENDNOW
C37	NPI/TON	NPI/TON

10.3.46 LCR DIMSU's

C40	LCR Route Elements	LCR-Richtungselemente
C41	LCR Digit Pattern	LCR-Wählmuster
C42	LCR Outdial Rules	LCR-Wahlregeln
C43	LCR Route Numbers	LCR-Richtungsnummern

10.3.47 V3.5 lcr, lcr_attribute

*		
C45	SUPPCPN	KANZATLN
C46	VOICO	VOICO

10.3.48 V3.6 lcr, lcr_attribute

C47	TGPUB	TGPUB
C48	TGPRIV	TGPRIV

CMI Values

C50	The first 5	die ersten 5
C51	The last 5	die letzten 5
C52	Layout 1	Anordnung 1
C53	Layout 2	Anordnung 2
C54	Inactive	nicht aktiv
C55	Active	Aktiv
C56	Activate	Aktivieren
C57	Activate and reset	aktivieren und zurücksetzen
C58	Activate statistics counter	aktivieren Statistikzähler
C59	verify	prüfen

CMI devtypes

C60	RADIO	RADIO
C61	BASEST	BASEST
C62	CORDLESS_E_ADM	CORDLESS_E_ADM
C63	EXTLINE	EXTLINE

CMI exact dev types

C64	SMT	SMT
C65	KMT	KMT

*

C66	Cordless E PIN	Cordless E PIN
C67	Cordless E STN	Cordless E TLN

*

C68	Deactivate statistics counter	Statistikzähler deaktivieren
C69	Not registered	nicht registriert
C70	Registered	registriert

*

C71	piktogram	piktogramm

Languages

C72	US	US

* C73 German/deutsch (duplicate of 6)

C74	english	englisch (international)
C75	Spanish	spanisch
C76	Portuguese	portugiesisch
C77	Scandinavian	skandinavisch
C78	Italian	italienisch
C79	Greek	griechisch

* C80 franzoesisch (duplicate of ???)

C81	French	französisch
C82	Finnish	finnisch
C83	Norwegian	norwegisch
C84	Swedish	schwedisch
C85	Czech	tschechisch
C86	Hungarian	ungarisch
C87	Polish	polnisch
C88	Danish	dänisch
C89	Dutch	niederländisch
C90	Portuguese (Brazilian)	portugiesisch (Brasilien)

10.3.49 CMI exact dev types

C91	GAP	GAP

10.3.50 hopping mode

C92	no hopping	no hopping
C93	slow hopping	slow hopping
C94	fast hopping	fast hopping

C95	Turkish	türkisch

10.3.51 V3.5 Values

* Huntgroup Values

* D00 Rufnummer	(duplicate of 614)	(duplicate of 614)
D01	Internal VoiceMail	interne Voicemail
D02	External VoiceMail	externe Voicemail
D03	Phone Mail	Phone Mail
* D04 Neben-stelle	(duplicate of D10)	(duplicate of D10)
* D05 Sammel-anschluß	(duplicate of D11)	(duplicate of D11)

D10	Station No. of Hunting Group Member	Rufnummer des SA-Teilnehmers	
D11	Extension No. of Hunt Group	Rufnummer von Sammelanschluss	
* D12	VOICE	VOICE	(duplicate of i90)
D13	DATA	DATEN	
* D14	FAX	FAX	(duplicate of 212)

10.3.52 Device types for Unity

D50	OPTIP500	OPTIP500	* optipoint phone

exact device types

D51	optiPoint 500 Entry	optiPoint 500 Entry
D52	optiPoint 500 Basic	optiPoint 500 Basic
D53	optiPoint 500 Standard	optiPoint 500 Standard
D54	optiPoint 500 Speaker U.S.	optiPoint 500 Speaker U.S.
D55	optiPoint 500 Advance	optiPoint 500 Advance
D56	optiPoint 600 Office	optiPoint 600 Office

D57	optiPoint 500 Economy	optiPoint 500 Economy
D58	optiPoint 400 Standard	optiPoint 400 Standard
D59	optiPoint 400 Economy	optiPoint 400 Economy

D60	OPTI	OPTI
D61	OPTI_T19	OPTI_T19
D62	optiClient 130	optiClient 130

D63	optiPoint 410 Entry	optiPoint 410 Entry
D64	optiPoint 410 Economy	optiPoint 410 Economy
D65	optiPoint 410 Standard	optiPoint 410 Standard
D66	optiPoint 410 Advance SLK	optiPoint 410 Advance SLK
D68	optiPoint 410 Economy SLK	optiPoint 410 Economy SLK
D69	optiPoint 410 Standard SLK	optiPoint 410 Standard SLK

D40	OPTI_T18	OPTI_T18
D90	optiPoint 410 economy plus	optiPoint 410 Economy Plus
D91	optiPoint 420 Economy Plus SLK	optiPoint 420 Economy Plus SLK
D95	optiPoint 420 Standard WLAN	optiPoint 420 Standard WLAN

D96	OPENSTAGE 20	OPENSTAGE 20
D97	OPENSTAGE 40	OPENSTAGE 40
D98	OPENSTAGE 60	OPENSTAGE 60
D99	OPENSTAGE 80	OPENSTAGE 80
D85	OPENSTAGE 10	OPENSTAGE 10
D86	OPENSTAGE 20E	OPENSTAGE 20E
D87	OPENSTAGE 15	OPENSTAGE 15
E85	OPENSTAGE 30	OPENSTAGE 30
E86	OPENSTAGE 30 BLF	OPENSTAGE 30 BLF
E87	OPENSTAGE 40 BLF	OPENSTAGE 40 BLF
E88	OPENSTAGE 60 BLF	OPENSTAGE 60BLF
E89	OPENSTAGE 80 BLF	OPENSTAGE 80 BLF
E90	OPENSTAGE 60_80	OPENSTAGE 60_80

E91	OPENSTAGE 60_80 BLF	OPENSTAGE 60_80 BLF

E92	OPTIDPI35	OpenScape Desk Phone IP 35
E93	OPTIDPI55	OpenScape Desk Phone IP 55
E94	OPTIDPI35 ECO	OpenScape Desk Phone IP 35 ECO
E95	UFA	UFA
E96	CP200	OpenScape Desk Phone CP200
E97	CP400	OpenScape Desk Phone CP400
E98	CP600	OpenScape Desk Phone CP600
E99	CP200T	OpenScape Desk Phone CP200 TDM
E60	CP400T	OpenScape Desk Phone CP400 TDM
E61	CP100	OpenScape Desk Phone DeskPhone CP100
E62	CP700	OpenScape Desk Phone DeskPhone CP700

CMI-Values

D70	Europe	Europa
D71	Latin America	Lateinamerika

*

D74	24 dBm	24 dBm
D75	4 dBm	4 dBm

*

D77	default protocol	Standardprotokoll
D78	VMS protocol	VMS-Protokoll
D79	SMS protocol	SMS-Protokoll
D80	DTB protocol	DTB-Protokoll
D81	MCL protocol	MCL-Protokoll

*

D82	activate BS	BS aktivieren
D83	deactivate BS	BS deaktivieren
D84	reset BS	BS rücksetzen

10.3.53 *Values for dual_mode_skey

E00	No MTP	Kein KMT
E01	Upgrade not started yet	Upgrade noch nicht gestartet
E02	Upgrade started	Upgrade gestartet
E03	Upgrade failed	Upgrade fehlerhaft
E04	set	eingestellt
E05	No support	keine Unterstützung

10.3.54 Values for dtb_access

E08	No access	kein Zugriff
E09	Access with call log	Zugriff mit Rufjournal
E10	Access without call log	Zugriff ohne Rufjournal

10.3.55 Values for version in cmi_board

E13	Cordless E V2.1	Cordless E V2.1
E14	Cordless E V2.2	Cordless E V2.2
E15	Cordless E V2.3	Cordless E V2.3
E16	Cordless E V2.4	Cordless E V2.4

10.3.56 Values for ic_data_type

E20	CMI static voice	CMI static voice
E21	CMI static data with voice band	CMI static data with voice band
E22	CMI static transparent data	CMI static transparent data
E23	CMI dynamic service recognition	CMI dynamic service recognition

10.3.57 V3.5 LCR-Values

* Values for Rule Types NPI und TON* attribute parameter in table lcrrule_sub

E30	ISDN/UNKNOWN	ISDN/UNKNOWN
E31	ISDN/SUBSCR	ISDN/SUBSCR

E32	ISDN/NATIONAL	ISDN/NATIONAL
E33	ISDN/INTERNAT	ISDN/INTERNAT
E34	PRIVATE/UNKNOWN	PRIVATE/UNKNOWN
E35	PRIVATE/LOCAL	PRIVATE/LOCAL
E36	PRIVATE/LEVEL1	PRIVATE/LEVEL1
E37	PRIVATE/LEVEL2	PRIVATE/LEVEL2
E38	UNKNOWN/UNKNOWN	UNKNOWN/UNKNOWN
E39	DATA/UNKNOWN	DATA/UNKNOWN
E40	LANDMOB/UNKNOWN	LANDMOB/UNKNOWN
E41	MARITMOB/UNKNOWN	MARITMOB/UNKNOWN
E42	NATIONAL/UNKNOWN	NATIONAL/UNKNOWN
E43	TELEX/UNKNOWN	TELEX/UNKNOWN

10.3.58 V3.6 LCR Values

*

10.3.59 lrtg, mfvpuls

F00	PP80	PP80
F01	PP150	PP150
F02	PP200	PP200
F03	PP300	PP300

10.3.60 DIMSU

F10	Number of HG3550 Boards	Anzahl HG3550-Baugruppen
F11	Number of registered gateways	Anzahl registrierter Gateways
F12	Number of path descriptions	Anzahl Beschreibungen der Verbindungswege
F13	Number of dialplan entries for LCR in pool 2	Anzahl Wahlberechtigungseinträge für LCR im Pool 2
F14	Number of entries in the LCR dial plan cache for pool 2	Anzahl Einträge im LCR-Wahlberechtigungs-Cache für Pool 2

F15	Number of dialplan entries for LCR in pool 3	Anzahl Wahlberechtigungseinträge für LCR im Pool 3
F16	Number of entries in the LCR dial plan cache for pool 3	Anzahl Einträge im LCR-Wahlberechtigungs-Cache für Pool 3
F17	Number of dialplan entries for LCR in pool 4	Anzahl Wahlberechtigungseinträge für LCR im Pool 4
F18	Number of entries in the LCR dial plan cache for pool 4	Anzahl Einträge im LCR-Wahlberechtigungs-Cache für Pool 4
F19	Number of dialplan entries for LCR in pool 5	Anzahl Wahlberechtigungseinträge für LCR im Pool 5
F20	Number of entries in the LCR dial plan cache for pool 5	Anzahl Einträge im LCR-Wahlberechtigungs-Cache für Pool 5
F21	Number of dialplan entries for LCR in pool 6	Anzahl Wahlberechtigungseinträge für LCR im Pool 6
F22	Number of entries in the LCR dial plan cache for pool 6	Anzahl Einträge im LCR-Wahlberechtigungs-Cache für Pool 6
F23	Number of dialplan entries for LCR in pool 7	Anzahl Wahlberechtigungseinträge für LCR im Pool 7
F24	Number of entries in the LCR dial plan cache for pool 7	Anzahl Einträge im LCR-Wahlberechtigungs-Cache für Pool 7
F25	Number of dialplan entries for LCR in pool 8	Anzahl Wahlberechtigungseinträge für LCR im Pool 8
F26	Number of entries in the LCR dial plan cache for pool 8	Anzahl Einträge im LCR-Wahlberechtigungs-Cache für Pool 8

10.3.61 pin,pin_type

G01	Business PIN long	PIN Dienstgespräch lang	
G02	Business PIN short	PIN Dienstgespräch kurz	
G03	PIN Private Call	PIN Privatgespräch	
G04	Project Code PIN	Projektcode-PIN	
G05	PIN for Service ID	Wartungskennzahl	
G06	PIN for Dialing Pattern	PIN für Wählmuster	* copin 10
G07	CDR Account Code	GE-Abrechnungscode	* copin 9

10.3.62 language,switch/port

H01	Estonian	estnisch
H02	Indonesian	indonesisch
H03	Catalan	katalanisch
H04	Croatian	kroatisch
H05	Malaysian	malaysisch
H06	Romanian	rumänisch
H07	Russian Cyrillic	russisch kyrillisch
H08	Slovakian	slowakisch
H09	Thai	thailändisch
H10	Slovenian	slowenisch
H11	Russian Latin	russisch lateinisch
H12	Chinese	chinesisch
H13	Latvian	lettisch
H14	Lithuanian	litauisch
H15	japanese	japanisch

10.3.63 Vermittlungsplatz Rufnummer

H20	ATNDIND	VPI

out_line_sel

H30	Primary	primär
H31	Last	letzte
H32	Free	freie

inc_line_sel

H33	Caller	rufende
H34	Caller (Primary line)	rufende (Primärleitung)
H35	all	alle
H36	all (Primary Line)	alle (Primärleitung)

ring_type

H37	No ringing	kein Ruf
H38	Warn ringing	Warnruf
H39	Rollover tone	Rollover-Ton

10.3.64 Flex-Shelf-Values

I00	FLEX	FLEX
I01	LOCAL	LOKAL

10.3.65 fix call forward

J01	CFU&CFB	CFU&CFB
J02	CFU&CFNR	CFU&CFNR
J03	CFU&CD	CFU&CD
J04	CFB&CFNR	CFB&CFNR
J05	CFB&CD	CFB&CD
J06	CFNR&CD	CFNR&CD
J07	CFU&CFB&CFNR	CFU&CFB&CFNR
J08	CFU&CFB&CD	CFU&CFB&CD
J09	CFNR&CD&CFB	CFNR&CD&CFB
J10	CFU&CD&CFNR	CFU&CD&CFNR
J11	CFU&CFB&CFNR&CD	CFU&CFB&CFNR&CD

10.3.66 Values for OpenScape Xpressions (Xpressions 450)

K00	Available	vorhanden
K01	Unavailable	nicht vorhanden

*

K10	Internal	intern
K11	local	lokal
K12	National	national
K13	International	international

10.3.67 special_subscriber, line_dist

K20	Less than 100	Kleiner als 100
K21	Between 100 and 1000	Zwischen 100 und 1000
K22	Greater than 1000	Größer als 1000

10.3.68 port,classmrk

K25	EC	EC	* Echo Cancellation
* K26	None	None	(duplicate of k54)
K27	VCG729A	VCG729A	* Voice Compression via G729A
K28	EC & VCG729A	EC & VCG729A	* Echo Cancellation & Voice Compression

10.3.69 display_name,charcon

K35	neutral	neutral
K36	russcyr	russcyr

10.3.70 again Object_type for action_protocol

O10	admin	admin
O11	adm_unique_key	adm_unique_key
O12	appearance	appearance
O13	appearance_on	appearance_on
O14	atm_board	atm_board
O15	Active COS Groups	Aktive COS-Gruppen
O16	System Networking	Anlagenvernetzung
O17	Building	Gebäude
* O18	Standard Key Layout	Standard Key Layout
O19	Call Forwarding	Anrufumleitung

O20	Coded Values	Kodierte Werte
O21	Chese Group	Chef Sekretär Gruppe
* O22	Networkwide Team	Networkwide Team
O23	Global Cordless E Data	Global Cordless E Daten
O24	Cordless E System Data	Cordless E Systemdaten
O26	Cordless Tie	Cordless-Querwahl
O27	col_domain	col_domain
O28	col_persdat	col_persdat
O29	col_pin	col_pin
O30	col_switch	col_switch
O31	Communication Group	Kommunikationsgruppe
O32	Company Code	Firmenkennzeichen
O33	conv_config	conv_config
* O34	Class of Service	Class of Service
O35	COS Switchover	BER-Umschaltung
O36	Call Progress States	Verkehrssituationen
O37	Country Table	Ländertabelle
O38	datadict	datadict
O39	Device Combination.	Gerätekombination.
* O40	Device Combination	Device Combination
O41	Device Combination/Key Layout	Geräte Kombination/Tastenbelegung
O42	Dial Code	Wählcode
O43	Dimension Data (dimsu)	Dimensionierungs-Daten (dimsu)
O44	Domain	Domain
O45	Download Class	Download-Klasse
O46	Download Index	Download-Index
O47	Download Parameter	Download-Parameter
O48	download_supp	download_supp
O49	dynamic	dynamic
O50	dynamic_keydat	dynamic_keydat
O51	evm_access	evm_access

* O52	Company	Company
O53	global	global
O54	Hotline	Hotline
O55	Hunt Group	Sammelanschluß
O56	Hunt Group Access Group	Sammelanschluß Zugriffscode
O57	Hunt Group (DPLN, CPS)	Sammelanschluß (WABE, VKS)
O58	Hunt Group Service	Sammelanschluß Dienst
O59	Node Format Tables (KNFOR)	Knotenformattabellen (KNFOR)
O60	Source Dep. Routing (KNLCR)	Ursprungsabh. Routing (KNLCR)
O61	Network Node Data (KNMAT)	Netzwerkknoten-Daten (KNMAT)
O62	Node Prefix Tables (KNPRE)	Knotenpräfix-Tabellen (KNPRE)
O63	Speed Dial	Kurzwahl
* O64	Speed Dial List	Speed Dial List
O65	LCR Authorizations (lber)	LCR-Berechtigungen (lber)
O66	LCR Authorizations (lcrauth)	LCR-Berechtigungen (lcrauth)
* O67	LCR Class of Service - Data	LCR Class of Service - Data
* O68	LCR Class of Service - Voice	LCR Class of Service - Voice
* O69	LCR Route Element	LCR Route Element
O70	LCR Overview	LCR-Ãbersicht
O71	Digit Pattern	Wählmuster
O72	Outdial Rule	Wahlregel
O73	Timetable	Zeitplan
O74	Zone Data	Zonendaten
O75	LCR Digit String Restriction Table	LCR-Digits String Einschränkungstabelle
* O76	Location	Location
* O77	LCR Route	LCR Route
O78	Night Options	Nachtvarianten
O79	Virtual Nodes (KNDEF)	Virtuelle Knoten (KNDEF)
O80	Order State	Auftragszustand
O81	Organization	Organisation

O82	PEN	Lage
* O83	Subscriber	Subscriber
* O84	Call Pickup	Call Pickup
O85	pickup_link	pickup_link
O86	PIN	PIN
O87	Class of PIN (COPIN)	PIN-Klasse (COPIN)
O88	General PIN Data	Allgemeine PIN-Daten
O89	PIN Check	PIN-Ãberprüfung
O90	Station.	Teilnehmeranschluß.
O91	port_model	port_model
O92	prime_line_key	prime_line_key
O93	Dial Numbers	Rufnummern
O94	Dialcodes (DPLN + CPS)	Rufnummern (Wabe + VKS)
O95	Dial Number Ranges	Rufnummernbereich
O96	refta	refta
O97	reminder	reminder
O98	Route info (Subscriber)	Richtungnsinformation (Teilnehmer)
O99	Route info (Net)	Richtungnsinformation (Netz)
P00	Number Conversion (RUFUM)	Rufnummernumsetzung (RUFUM)
P01	Scheduled Batch	Zeitauftrag
P02	Shelf Data	BGR Daten
P03	Special Stations	Sonderteilnehmer
P04	Speed Dial List.	Kurzrufnummernliste.
P05	station_xref	station_xref
P06	System Default	Vorbelegung pro Anlage
P08	Templates	Muster
P09	Network Dial Plan (TWABE)	Netzwählplan (TWABE)
P10	upd_persdat	upd_persdat
P11	upd_pin	upd_pin
P12	upd_port	upd_port
P13	User Default	Vorbelegung pro Benutzer

P14	Traffic Restriction	Verkehrsbeziehung
P15	Internal Traffic Restrictions Card	Verkehrsbeziehung Ausweis
P16	Traffic Restriction Group	Verkehrsbeziehungsgruppe
P17	VoiceMail COS	VoiceMail COS
P18	Voice-Mail	VoiceMail

P30	Individual Key Layout.	individuelle Tastenbelegung
P31	Individual Key Layout 2	individuelle Tastenbelegung 2
P32	Individual Key Layout 3	individuelle Tastenbelegung 3
P33	Individual Key Layout 4	individuelle Tastenbelegung 4
P34	Traffic Restriction Group.	Verkehrsbeziehungsgruppe.
P35	Consistency	Konsistenz
P36	Consistency Info	Konsistenz Info
P37	conv_key	conv_key
P38	Dimension Data List	Dimensionierungs-Daten Liste
P39	Home Location Codes	Home Location Codes
P40	Hunt Group Data.	Daten Sammelanschluß.
P41	Hunt Group Data .	Daten Sammelanschluß .
P43	Class of Service.	Berechtigungsklasse.
P44	h_cosxt	h_cosxt
P45	h_devtyp_rel	h_devtyp_rel
P46	h_huntgrp	h_huntgrp
P47	h_night_opt	h_night_opt
P48	h_pen	h_pen
P49	h_pickupgrp	h_pickupgrp
P50	h_pin	h_pin
P51	h_port	h_port
P52	h_shelf_data	h_shelf_data
P53	icom	icom
P54	keydat	keydat
P55	lcos	lcos
* P56	PhoneMail Mailbox	PhoneMail Mailbox

P57	pmnode	pmnode
* P58	PhoneMail System	PhoneMail System
P59	pm_interface	pm_interface
P60	Port Configuration	Anschluß Konfiguration
P61	Port Data 1	Anschluß
P62	Port Data 2	Anschluß
P63	Port Data 3	Anschluß
P64	Port Data 4	Anschluß
P65	Port Fax	Anschluß Fax
P66	remote	remote
P67	Multi View	Mandantenverwaltung
P68	Route Index.	Richtungsindex.
P69	Route Info	Richtungsinfo
P70	r_button_std	r_button_std
P71	r_button_indiv	r_button_indiv
P72	r_cos	r_cos
P73	r_cosxt	r_cosxt
P74	r_devtyp_rel	r_devtyp_rel
P75	r_devconst_sub	r_devconst_sub
P76	r_pen	r_pen
P77	r_pickupgrp	r_pickupgrp
P78	r_port	r_port
P79	r_shelf_data	r_shelf_data
P80	serv_interface	serv_interface
P81	System Speed Dial Lists.	Zentralkurzwahlliste.
P82	stnum_reference	stnum_reference
P83	Converged Standard Key Layout View	Ansicht aller Standard-Tastenbelegungen
P84	Converged COS View	Ansicht aller COS
P85	Converged COS Switchover View	Ansicht aller COS-Umschaltungen
P86	sv_device	sv_device

P87	Converged Hunt Group View	Ansicht aller Sammelanschlüsse
P88	sv_lcos	sv_lcos
P89	Converged PEN View	Ansicht aller Lagen
P90	Converged Pickup Group View	Ansicht aller Anrufübernahmegruppen
P91	Converged PIN View	Ansicht aller PIN
P92	Voice Station Move	Umzug des Sprach-Teilnehmers
P93	Converged Speed Dial View	Ansicht aller Kurzwahlen
P94	syscon	syscon
P95	twclient	twclient
P96	used_butt_std	used_butt_std
P97	xpupload	xpupload
Q02	LCR Profiles	LCR Profile
Q03	Source Group	Quellgruppe
Q04	External VoiceMail Service	VoiceMail Service extern

10.3.71 Special values

Z98	ILLEGAL	UNGÃLTIG	*undefined/out-of-range for DBSYNC
* Z99 ?	*this one can't be used, because it's used in the code!!	*this one can't be used, because it's used in the code!!	

10.3.72 ** section for subdomain codes (also another section is at the end)

Values for LCT-API_PartNumber

a00	Q2031-X4	Q2031-X4
a01	Q2052-X	Q2052-X
a02	Q2075-X	Q2075-X
a03	Q2086-X	Q2086-X
a04	Q2092-X	Q2092-X
a05	Q2098-X	Q2098-X
a06	Q2100-XBOS	Q2100-XBOS

a07	Q2100-XMOS	Q2100-XMOS
a08	Q2117-X	Q2117-X
a09	Q2128-X	Q2128-X
a10	Q2133-X	Q2133-X
a11	Q2133-X100	Q2133-X100
a12	Q2136-X	Q2136-X
a13	Q2139-X	Q2139-X
a14	Q2146-X	Q2146-X
a15	Q2148-X	Q2148-X
a16	Q2150-X	Q2150-X
a17	Q2150-X100	Q2150-X100
a18	Q2158-X	Q2158-X
a19	Q2160-X	Q2160-X
a20	Q2160-X100	Q2160-X100
a21	Q2161-X	Q2161-X
a22	Q2197-X	Q2197-X
a23	Q2205-X	Q2205-X
a24	Q2208-X200	Q2208-X200
a25	Q2233-X	Q2233-X
a26	Q2246-X	Q2246-X
a27	Q2248-X	Q2248-X
a28	Q2271-X	Q2271-X
a29	Q2450-X	Q2450-X
a30	Q2451-X	Q2451-X
a31	Q2452-X	Q2452-X
a32	Q2467-X	Q2467-X
a33	Q2468-X	Q2468-X
a34	Q2469-X	Q2469-X
a35	Q2471-XBOS	Q2471-XBOS
a36	Q2471-XMOS	Q2471-XMOS
a37	Q2472-X	Q2472-X
a38	Q2473-X	Q2473-X

a39	Q2474-X	Q2474-X	
a40	Q2474-X100	Q2474-X100	
a41	Q2475-X	Q2475-X	
a42	Q2479-X	Q2479-X	
a43	Q2484-X	Q2484-X	
a44	Q2485-X	Q2485-X	

Values for LCT-API_PublicNumType

*	Unknown	Unknown	(duplicate of)
a46	Not Defined	Nicht definiert	
*	International	International	(duplicate of)
*	National	National	(duplicate of)
a47	Network	Netzwerk	
*	Subscriber	Subscriber	(duplicate of)
*	Reserved	Reserved	(duplicate of)
a48	Abbreviated	Kurzwahl_	
a49	Regional Level 1	Regional-Ebene 1	
a50	Regional Level 2	Regional-Ebene 2	
*	Local	Local	(duplicate of)

Values for LCT-API_QtrShelf

a52	QTR_SHELF_1	QTR_SHELF_1	
a53	QTR_SHELF_2	QTR_SHELF_2	
a54	QTR_SHELF_3	QTR_SHELF_3	
a55	QTR_SHELF_4	QTR_SHELF_4	

Values for LCT-API_RMS_Status

a57	NO	NO	
*	V1	V1	(duplicate of)
*	V2	V2	(duplicate of)
a58	NOT_ALLOWED	NOT_ALLOWED	

Values for LCT-API_RangeType

a60	PEN_RANGE	PEN_RANGE	

a61	STATION_RANGE	STATION_RANGE
a62	ACCESS_RANGE	ACCESS_RANGE
a63	WORD_RANGE	WORD_RANGE

Values for LCT-API_RecorderApplication

a65	SYSTEM_MUSIC	SYSTEM_MUSIC
a66	DID_INTERCEPT_ANNOUNCEMENT	DID_INTERCEPT_ANNOUNCEMENT
a67	ACD_ANNOUNCEMENT	ACD_ANNOUNCEMENT
a68	ACD_MUSIC	ACD_MUSIC

Values for LCT-API_RingOption

*	NO	NO	(duplicate of)
a70	YES	YES	

Values for LCT-API_RingType

a72	Abbreviated Ring	Kurzruf
a73	Standard Ring	Standard Ruf
a74	No Ring	Kein Ruf

Values for LCT-API_RoImPhoneType

*	RP120	RP120	(duplicate of)
*	RP240	RP240	(duplicate of)
a76	RP240B	RP240B	
a77	RP240E	RP240E	
*	RP312	RP312	(duplicate of)
a78	RP312L	RP312L	
*	RP400	RP400	(duplicate of)
*	RP612	RP612	(duplicate of)
a79	RP612L	RP612L	
a80	RP612S	RP612S	
a81	RP612SL	RP612SL	
a82	RP612D	RP612D	
a83	RP612SD	RP612SD	
a84	RP612LD	RP612LD	

a85	RP612SLD	RP612SLD	
*	RP612K	RP612K	(duplicate of)
a86	RP612SK	RP612SK	
a87	RP612LK	RP612LK	
a88	RP612SLK	RP612SLK	
a89	RP612DK	RP612DK	
a90	RP612SDK	RP612SDK	
a91	RP612LDK	RP612LDK	
a92	RP612SLDK	RP612SLDK	
*	RP624	RP624	(duplicate of)
a93	RP624L	RP624L	
a94	RP624S	RP624S	
a95	RP624SL	RP624SL	
a96	RP624D	RP624D	
a97	RP624SD	RP624SD	
a98	RP624LD	RP624LD	
a99	RP624SLD	RP624SLD	
*	RP624K	RP624K	(duplicate of)
b00	RP624SK	RP624SK	
b01	RP624LK	RP624LK	
b02	RP624SLK	RP624SLK	
b03	RP624DK	RP624DK	
b04	RP624SDK	RP624SDK	
b05	RP624LDK	RP624LDK	
b06	RP624SLDK	RP624SLDK	

Values for LCT-API_ServiceIndicator

b08	Voice	Sprache
b09	Data	Daten
b10	Fax	Fax

Values for LCT-API_ShelfType

b12	TYPE_NULL	TYPE_NULL

*	NORMAL	NORMAL	(duplicate of)
*	COMPACT	COMPACT	(duplicate of)
b13	H_80CM_BASE	H_80CM_BASE	
b14	CC80X	CC80X	
b15	CC80F	CC80F	
b16	EXT_COMP_X	EXT_COMP_X	

Values for LCT-API_SignalingType

b18	CAS	CAS
b19	BOS	BOS
b20	CCO	CCO
b21	CVN	CVN
b22	MOS	MOS

Values for LCT-API_SpecialEquipment

b24	Null Equipment	Null-Einrichtung
b25	Special Station	Sonderteilnehmer_
b26	Dictation Equipment	Diktiereinrichtung
b27	Paging	Personensuche

Values for LCT-API_StopBits

b29	STOP_BITS_1	STOP_BITS_1
b30	STOP_BITS_1_PT_5	STOP_BITS_1_PT_5
b31	STOP_BITS_2	STOP_BITS_2

Values for LCT-API_SwitchoverCOS_Active

b32	DAY	DAY
b33	NIGHT	NNST
b34	LEAVE_THE_SAME	LEAVE_THE_SAME

Values for LCT-API_SystemSpeedDialSize

b35	_SHORT_100	_SHORT_100
b36	_LONG_1000	_LONG_1000

Values for LCT-API_SwitchoverCOS_Switch

b37	COSX_OFF	COSX_OFF	
b38	COSX_ON	COSX_ON	

Values for LCT-API_TestType

*	LOOPBACK	LOOPBACK	(duplicate of)
b40	SELFTEST	SELFTEST	
b41	CHECK_ALIVE	CHECK_ALIVE	
b42	LINE_POWER	LINE_POWER	

Values for LCT-API_TimingType

b43	SYST	SYST
b44	LOOP	LOOP

Values for LCT-API_TransmissionMode

b45	HALF	HALF
b46	FULL	FULL

Values for LCT-API_TransmissionType

b47	SYNC	SYNC
b48	ASYNC	ASYNC

Values for LCT-API_UbgDat

b50	SACHNR	SACHNR
b51	PART_NUMBER	PART_NUMBER
b52	HW_ABKUERZ	HW_ABKUERZ
b53	BOARD_NAME	BOARD_NAME
b54	KENNUNG	KENNUNG
b55	HARDWARE_ID	HARDWARE_ID
b56	CARD_SLOT_TYP	CARD_SLOT_TYP
b57	AM_BOARD_TYP	AM_BOARD_TYP
b58	FA_BOARD_TYP	FA_BOARD_TYP
b59	DC_BOARD_TYP	DC_BOARD_TYP
b60	RTO_BOARD_TYP	RTO_BOARD_TYP
b61	NMC_BOARD_TYP	NMC_BOARD_TYP

b62	REC_BOARD_TYP	REC_BOARD_TYP
b63	PP_BOARD_TYP	PP_BOARD_TYP
b64	BG_LADBAR	BG_LADBAR
b65	ANZ_SAETZE	ANZ_SAETZE
b66	ANZ_CS	ANZ_CS
b67	LW_FORMAT	LW_FORMAT
b68	MP_BLOCK_LENGTH	MP_BLOCK_LENGTH
b69	ERW_DATA_VORH	ERW_DATA_VORH
b70	FCT_ID	FCT_ID
b71	FNCT_I	FNCT_I
b72	AM_LINE_TYP	AM_LINE_TYP
b73	FA_LINE_TYP	FA_LINE_TYP
b74	DC_LINE_TYP	DC_LINE_TYP
b75	RTO_LINE_TYP	RTO_LINE_TYP
b76	NMC_LINE_TYP	NMC_LINE_TYP
b77	REC_LINE_TYP	REC_LINE_TYP
b78	PP_LINE_TYP	PP_LINE_TYP
b79	PIT_FCT_TYP	PIT_FCT_TYP
b80	FILE_NAME	FILE_NAME
b81	BOARD_DATA	BOARD_DATA
b82	MIN_BG_VERS	MIN_BG_VERS
b83	MIN_FW_VERS	MIN_FW_VERS
b84	ANZ_CS_COMPACT	ANZ_CS_COMPACT
b85	FLASH_HANDLING	FLASH_HANDLING
b86	FLASH_DEV	FLASH_DEV
b87	RAM_BLOCK_SIZE	RAM_BLOCK_SIZE
b88	TIME_DEL_FLASH	TIME_DEL_FLASH
b89	TIME_WRITE_KBYTE	TIME_WRITE_KBYTE
b90	RESTART_ABILITY	RESTART_ABILITY
b91	SELTEST_TIME	SELTEST_TIME
b92	LOADWARE	LOADWARE
b93	LOAD_PRIORITY	LOAD_PRIORITY

b94	DIAGNOSIS_FCT	DIAGNOSIS_FCT
b95	HW_PLATFORM	HW_PLATFORM
b96	FLOW_CTRL_POSSIBLE	FLOW_CTRL_POSSIBLE
b97	SHORT_TERM_VALUE	SHORT_TERM_VALUE
b98	LONG_TERM_VALUE	LONG_TERM_VALUE
b99	PERM_TERM_VALUE	PERM_TERM_VALUE

Values for LCT-API_Unit

c02	LTG	LTG	
c03	LTU	LTU	
c04	BRD	BRD	
c05	CIR	ZYK	
*	CTE	CTE	(duplicate of)
c06	EG	EG	
*	CONF	CONF	(duplicate of)
*	SIU	SIU	(duplicate of)
c07	SIP	SIP	
c08	SIU_FKT	SIU_FKT	
c09	HPBC	HPBC	
*	IP	IP	(duplicate of)
c10	CCH	CCH	
*	DP	DP	(duplicate of)
c11	DEV	DEV	
c12	SM	SM	
*	DCL	DCL	(duplicate of)
c13	HDLC	HDLC	
c14	GSN	GSN	
*	MTS	MTS	(duplicate of)
c15	HWY	HWY	
c16	TSL	TSL	
*	IOCG	IOCG	(duplicate of)
c17	MEM	MEM	

c18	CSN	CSN	
*	IOCC	IOCC	(duplicate of)
*	CCG	CCG	(duplicate of)
*	CC	CC	(duplicate of)
*	MBU	MBU	(duplicate of)
c19	RGEN	RGEN	
c20	WGEN	WGEN	
c21	MEM_L	MEM_L	
c22	MEM_C	MEM_C	
*	MD	MD	(duplicate of)
c23	CLOK_REF	CLOK_REF	
c24	XENIX	XENIX	
c25	PLL	PLL	
*	VC	VC	(duplicate of)

Values for LCT-API_CORNET_TS_CUI

c28	FALSE	FALSE
c29	TRUE	TRUE

Values for LCT-API_OptisetE_API

c31	OPTISETE_API_FALSE	OPTISETE_API_FALSE
c32	OPTISETE_API_TRUE	OPTISETE_API_TRUE

Values for LCT-API_OptisetE_TAType

*	Data	Data	(duplicate of b09)
*	Analog	Analog	(duplicate of s10)
c34	Control	Steuerung	
*	ISDN	ISDN	(duplicate of p57)
c35	Reserved Data	Reserviert Daten	
c36	Reserved Analog	Reserviert Analog	

Values for LCT-API_PEN

*	UNKNOWN	UNKNOWN	(duplicate of)

Values for LCT-API_TA_UsageType

c39	CORNETT	CORNETT	
c40	Multifunc Terminal Adapter	Multifunc-Terminal-Adapter	
c41	NI Data	NI-Daten	
*	DCM	DCM	(duplicate of)
*	DCO	DCO	(duplicate of)
c42	CCP DCM	CCP DCM	
c43	DCM Modem Pool	DCM-Modem-Pool	
c44	Modem Pool	Modem-Pool	
c45	CCP DCI	CCP DCI	
c46	DCI AVD	DCI AVD	

Values for Board Funct-ID

c47	0 (Default)	0 (Standard)	
c48	1 (Default)	1 (Standard)	
c49	2 (Default)	2 (Standard)	
c50	3 (Default)	3 (Standard)	
c51	4 (Default)	4 (Standard)	
c52	TMDN CAS	TMDN CAS	
c53	TMDN CCO	TMDN CCO	
c54	TMDN CVN	TMDN CVN	
c55	TMDN MOS	TMDN MOS	
c56	SIUX PER	SIUX PER	
c57	SIUX TDS	SIUX TDS	
c58	STMA IW	STMA IW	
c59	STMA PBB	STMA PBB	

Values for Shelf Sub-Type

*	Local	Local	(duplicate of)
*	FLEX	FLEX	(duplicate of)

Values for LCT-API_MemDimenType

c60	ACDAG - CC2	ACDAG - CC2

c61	ACDCALEN - CC1	ACDCALEN - CC1
c62	ACDCQ - CC2	ACDCQ - CC2
c63	ACDEOS - CC2	ACDEOS - CC2
c64	ACDGR - CC2	ACDGR - CC2
c65	ACDLOGO - CC2	ACDLOGO - CC2
c66	ACDPRIM - CC2	ACDPRIM - CC2
c67	ACDRCG - CC2	ACDRCG - CC2
c68	ACDSHFT - CC2	ACDSHFT - CC2
c69	APILTG - CC1	APILTG - CC1
c70	APPL - CC1	APPL - CC1
c71	ART - CC2	ART - CC2
c72	ATND - CC1	ATND - CC1
c73	ATNDCQ - CC1	ATNDCQ - CC1
c74	ATNDGR - CC1	ATNDGR - CC1
c75	AULEXT - CC2	AULEXT - CC2
c76	AUTOCLL - CC2	AUTOCLL - CC2
c77	AUTOLTG - LTG	AUTOLTG - LTG
c78	AUXART - CC2	AUXART - CC2
c79	BCHNLSRT - CC1	BCHNLSRT - CC1
c80	CALLBK - CC2	CALLBK - CC2
c81	CCCEVENT - CC2	CCCEVENT - CC2
c82	CCIDXDYN - CC*	CCIDXDYN - CC*
c83	CCIDXSTA - CC*	CCIDXSTA - CC*
c84	COMCPB - CC1	COMCPB - CC1
c85	COMGR - CC2	COMGR - CC2
c86	COS - CC1	COS - CC1
c87	CPB - CC1	CPB - CC1
c88	CPDBUF - CC1	CPDBUF - CC1
c89	DATAGRP - CC2	DATAGRP - CC2
c90	DATAPOOL - CC2	DATAPOOL - CC2
c91	DCIPAR - CC1	DCIPAR - CC1
c92	DHBUFLTG - LTG	DHBUFLTG - LTG

c93	DIGADD - CC2	DIGADD - CC2
c94	DIGILTG - LTG	DIGILTG - LTG
c95	DILTG - LTG	DILTG - LTG
c96	DIBR3 - CC1	DIBR3 - CC1
c97	DIG2LTG - LTG	DIG2LTG - LTG
c98	DIRECT - CC1	DIRECT - CC1
c99	DISACODE - CC2	DISACODE - CC2
d00	DLINEPW - CC2	DLINEPW - CC2
d01	DLPWLTG - LTG	DLPWLTG - LTG
d02	DNI - CC2	DNI - CC2
d03	DNMBLK - CC2	DNMBLK - CC2
d04	DPICPD - CC2	DPICPD - CC2
d05	DPIDCIP - CC2	DPIDCIP - CC2
d06	DPIDNIT - CC2	DPIDNIT - CC2
d07	DPIKEYS - CC2	DPIKEYS - CC2
d08	DPISTMA - CC2	DPISTMA - CC2
d09	DPITAP - CC2	DPITAP - CC2
d10	DPITCSU - CC2	DPITCSU - CC2
d11	DPLNAC - CC1	DPLNAC - CC1
d12	DPLNGR - CC1	DPLNGR - CC1
d13	DPMCPD - CC2	DPMCPD - CC2
d14	DPMDCIP - CC2	DPMDCIP - CC2
d15	DPMDNIT - CC2	DPMDNIT - CC2
d16	DPMKEYS - CC2	DPMKEYS - CC2
d17	DPMSTMA - CC2	DPMSTMA - CC2
d18	DPMTAP - CC2	DPMTAP - CC2
d19	DPMTCSU - CC2	DPMTCSU - CC2
d20	DSLTG - LTG	DSLTG - LTG
d21	DTILTG - LTG	DTILTG - LTG
d22	RDTILTG - LTG	RDTILTG - LTG
d23	DSSPTNR - CC1	DSSPTNR - CC1
d24	DYAD - CC2	DYAD - CC2

d25	ECCS - CC1	ECCS - CC1
d26	EXTBLK - CC2	EXTBLK - CC2
d27	GDTR - CC2	GDTR - CC2
d28	GENBUF - CC1	GENBUF - CC1
d29	GRNAME - CC2	GRNAME - CC2
d30	HUNTCDB - CC1	HUNTCDB - CC1
d31	HUNTSTN - CC1	HUNTSTN - CC1
d32	IVDW - CC2	IVDW - CC2
d33	KEYSET - CC1	KEYSET - CC1
d34	KSETLTG - LTG	KSETLTG - LTG
d35	LAORT - CC1	LAORT - CC1
d36	LCB - CC2	LCB - CC2
d37	LODR - CC2	LODR - CC2
d38	LODS0 - CC2	LODS0 - CC2
d39	LDPLN - CC1	LDPLN - CC1
d40	LODEN - CC1	LODEN - CC1
d41	LROUT - CC1	LROUT - CC1
d42	LTGIDX - LTG	LTGIDX - LTG
d43	MDR - CC1	MDR - CC1
d44	MG - CC1	MG - CC1
d45	MONBUF - CC1	MONBUF - CC1
d46	MONID - CC2	MONID - CC2
d47	MWRESTRT - CC2	MWRESTRT - CC2
d48	NAME - CC1	NAME - CC1
d49	NSTN - CC1	NSTN - CC1
d50	NWCR - CC1	NWCR - CC1
d51	NWCRLTG - LTG	NWCRLTG - LTG
d52	OPTILTG - LTG	OPTILTG - LTG
d53	OPTIOPT - CC1	OPTIOPT - CC1
d54	OPTISET - CC1	OPTISET - CC1
d55	OPTSTLTG - LTG	OPTSTLTG - LTG
d56	ORG - CC1	ORG - CC1

d57	PERBG - CC2	PERBG - CC2
d58	PID - CC1	PID - CC1
d59	PRTRKLTG - LTG	PRTRKLTG - LTG
d60	PRITRUNK - CC2	PRITRUNK - CC2
d61	PUGR - CC1	PUGR - CC1
d62	DTIS0 - LTG	DTIS0 - LTG
d63	RLODEN - CC1	RLODEN - CC1
d64	RP120 - CC2	RP120 - CC2
d65	RP120LTG - LTG	RP120LTG - LTG
d66	RP240 - CC2	RP240 - CC2
d67	RP240LTG - LTG	RP240LTG - LTG
d68	RP400LTG - LTG	RP400LTG - LTG
d69	RP400PL - CC2	RP400PL - CC2
d70	RP4327 - CC2	RP4327 - CC2
d71	S0BUS - CC2	S0BUS - CC2
d72	S0LTG - LTG	S0LTG - LTG
d73	SG - CC1	SG - CC1
d74	SILPOOL - CC2	SILPOOL - CC2
d75	SLMPX - CC1	SLMPX - CC1
d76	SPDCD - CC1	SPDCD - CC1
d77	SPDCL - CC1	SPDCL - CC1
d78	SPDI - CC1	SPDI - CC1
d79	STMA - CC1	STMA - CC1
d80	STN - CC1	STN - CC1
d81	SU - CC1	SU - CC1
d82	T1SPAN - CC2	T1SPAN - CC2
d83	TFAGRP - CC1	TFAGRP - CC1
d84	TRUNK - CC1	TRUNK - CC1
d85	WSNAME - CC1	WSNAME - CC1

Values for LCT-API_StationNumRefTbl

d88	DNI	DNI

d89	ART	ART	
d90	ACD_SUPERVISOR	ACD_SUPERVISOR	
*	ACD_LOGON	ACD_LOGON	(duplicate of)
d91	NIGHT_OPT	NIGHT_OPT	
d92	NIGHT_VAR	NIGHT_VAR	
d93	ACL	ACL	

Values for LCT-API_DAR

*	STN	STN	(duplicate of)
d96	HUNT_GRP	HUNT_GRP	
d97	NOT_POSSIBLE	NOT_POSSIBLE	
*	DATA_DEST	DATA_DEST	(duplicate of)
d98	FAX_DEST	FAX_DEST	
d99	ATC_INDIVIDUAL_NUM	ATC_INDIVIDUAL_NUM	
e00	ATC_NUM_EXTERN	ATC_NUM_EXTERN	
e01	ATC_NUM_INTERN	ATC_NUM_INTERN	
e02	DIRECTION_NUM	DIRECTION_NUM	
e03	NETWORK_NUM	NETWORK_NUM	
e04	TIE_TRUNK1	TIE_TRUNK1	
e05	TIE_TRUNK2	TIE_TRUNK2	
e06	SYS_SPEED_DIAL1	SYS_SPEED_DIAL1	
e07	SYS_SPEED_DIAL2	SYS_SPEED_DIAL2	
e08	DNI_NUM	DNI_NUM	
e09	MODEM_POOL	MODEM_POOL	
e10	CODE_CONVERSION_POOL	CODE_CONVERSION_POOL	
e11	DCM_MODEM_POOL	DCM_MODEM_POOL	
e12	FAX_ACTIVATION	FAX_ACTIVATION	
e13	INTERNAL_FAX_SERVER	INTERNAL_FAX_SERVER	
e14	EXTERNAL_FAX_SERVER	EXTERNAL_FAX_SERVER	
e15	CALL_PICKUP	CALL_PICKUP	
e16	COM_GRP	COM_GRP	
e17	DIRECTED_CALL_PICKUP	DIRECTED_CALL_PICKUP	

e18	B_CHANNEL	B_CHANNEL	
e19	HUNT_GRP_VCE_ACT	HUNT_GRP_VCE_ACT	
e20	HUNT_GRP_VCE_DEA	HUNT_GRP_VCE_DEA	
e21	HUNT_GRP_FAX_ACT	HUNT_GRP_FAX_ACT	
e22	HUNT_GRP_FAX_DEA	HUNT_GRP_FAX_DEA	
e23	HUNT_GRP_DATA_ACT	HUNT_GRP_DATA_ACT	
e24	HUNT_GRP_DATA_DEA	HUNT_GRP_DATA_DEA	
e25	COXFER	COXFER	
*	SCONSI	SCONSI	(duplicate of)
*	SCONSCO	SCONSCO	(duplicate of)
e26	UNVO	UNVO	
e27	COS_CHGOVER_FOR_SUB_ACT	COS_CHGOVER_FOR_SUB_ACT	
e28	COS_CHGOVER_FOR_SUB_DEA	COS_CHGOVER_FOR_SUB_DEA	
e29	INDIV_SPEED_DIAL	INDIV_SPEED_DIAL	
e30	INDIV_SPEED_DIAL_PROG	INDIV_SPEED_DIAL_PROG	
e31	VCFW_NA_INT_EXT_ON	VCFW_NA_INT_EXT_ON	
e32	VCFW_BUSY_INT_EXT_ON	VCFW_BUSY_INT_EXT_ON	
e33	VCFW_NA_BUSY_INT_EXT_ON	VCFW_NA_BUSY_INT_EXT_ON	
e34	VCFW_ALL_INT_EXT_ON	VCFW_ALL_INT_EXT_ON	
e35	VCFW_ALL_INT_EXT_OFF	VCFW_ALL_INT_EXT_OFF	
e36	VCFW_ALL_INTERNAL_ON	VCFW_ALL_INTERNAL_ON	
e37	VCFW_ALL_EXTERNAL_ON	VCFW_ALL_EXTERNAL_ON	
e38	FCFW_ACTIVATED	FCFW_ACTIVATED	
e39	FCFW_CANCELLED	FCFW_CANCELLED	
e40	CAMP_ON	CAMP_ON	
e41	PU_UNIV_NIGHT_SERVICE	PU_UNIV_NIGHT_SERVICE	
e42	SAVED_NUMBER_REDIAL	SAVED_NUMBER_REDIAL	
e43	LAST_NUMBER_REDIAL	LAST_NUMBER_REDIAL	
e44	CALLBACK_ACT	CALLBACK_ACT	
e45	CALLBACK_DEA	CALLBACK_DEA	
e46	TOGGLE	TOGGLE	
e47	TIMED_REMINDER_ACT	TIMED_REMINDER_ACT	

e48	TIMED_REMINDER_DEA	TIMED_REMINDER_DEA
e49	LINE_TESTING_CIRCUIT	LINE_TESTING_CIRCUIT
e50	DO_NOT_DISTURB_ACT	DO_NOT_DISTURB_ACT
e51	DO_NOT_DISTURB_DEA	DO_NOT_DISTURB_DEA
e52	THIRD_PARTY_CONF	THIRD_PARTY_CONF
e53	DATA_AC	DATA_AC
e54	ENT_TELE_SYS_CHK	ENT_TELE_SYS_CHK
e55	ENT_TELE_SYS_OPEN	ENT_TELE_SYS_OPEN
e56	SPEC_RNG_ANNOUNCEMENT	SPEC_RNG_ANNOUNCEMENT
e57	CALL_ANSW_UNIT_CO	CALL_ANSW_UNIT_CO
e58	CALL_ANSW_UNIT_INTERNAL	CALL_ANSW_UNIT_INTERNAL
e59	DIGITAL_DEVICE_DATA	DIGITAL_DEVICE_DATA
* e60 TRACE	* (duplicate of o00)	* (duplicate of o00)
e61	UNVO_VKS	UNVO_VKS
e62	DCI_SPEED_CHG	DCI_SPEED_CHG
e63	DCI_CODE_RESET	DCI_CODE_RESET
e64	DCI_CODE_CHG	DCI_CODE_CHG
e65	PERSONAL_ID_ON	PERSONAL_ID_ON
e66	PERSONAL_ID_OFF	PERSONAL_ID_OFF
e67	BUSY_VER_BY_ATND	BUSY_VER_BY_ATND
e68	DIRECT_TRUNK_ACCESS	DIRECT_TRUNK_ACCESS
e69	STN_HUNT_PROG_AC	STN_HUNT_PROG_AC
e70	STN_HUNT_CLEAR_AC	STN_HUNT_CLEAR_AC
e71	MW_ACTIVATE	MW_ACTIVATE
e72	MW_CANCEL_BY_TERM_PTY	MW_CANCEL_BY_TERM_PTY
e73	MW_CANCEL_BY_ORIG_PTY	MW_CANCEL_BY_ORIG_PTY
e74	MW_ANSWER	MW_ANSWER
e75	DCI_LOC_TERM_LBACK_TEST	DCI_LOC_TERM_LBACK_TEST
e76	DCI_REM_CHNL_LBACK_TEST	DCI_REM_CHNL_LBACK_TEST
e77	DCI_REM_TERM_LBACK_TEST	DCI_REM_TERM_LBACK_TEST
e78	REMOVE_PTY_FROM_CONF	REMOVE_PTY_FROM_CONF

e79	CALL_HOLD	CALL_HOLD	
e80	SYSTEM_PARK	SYSTEM_PARK	
e81	VCE_DATA_SWITCHOVER	VCE_DATA_SWITCHOVER	
e82	DISA_DIRECTORY_NUM	DISA_DIRECTORY_NUM	
e83	REM_LAST_PTY_FROM_CONF	REM_LAST_PTY_FROM_CONF	
e84	PRI_TEST_AC	PRI_TEST_AC	
e85	DATA_56KBPS	DATA_56KBPS	
e86	RINGER_PITCH_SELECT	RINGER_PITCH_SELECT	
e87	INTERCOM_CALL	INTERCOM_CALL	
e88	BAD_LINE	BAD_LINE	
e89	SIMPLEX_VCE_CALL	SIMPLEX_VCE_CALL	
e90	PRIVACY_ON	PRIVACY_ON	
e91	PRIVACY_OFF	PRIVACY_OFF	
*	BUZZ	BUZZ	(duplicate of)
e92	CONSULTATION	CONSULTATION	
*	CLEAR	CLEAR	(duplicate of)
e93	CONFERENCE	CONFERENCE	
e94	ACCOUNT_CODE_DGT_COLLECT	ACCOUNT_CODE_DGT_COLLECT	
e95	ASR_RELOCATE_FROM	ASR_RELOCATE_FROM	
e96	ASR_RELOCATE_TO	ASR_RELOCATE_TO	
e97	ASR_3RD_PTY_RELOCATE	ASR_3RD_PTY_RELOCATE	
e98	PHONEMAIL_MW_CALLBACK	PHONEMAIL_MW_CALLBACK	
e99	PHONEMAIL_MW_CANCEL	PHONEMAIL_MW_CANCEL	
*	LOCAL	LOCAL	(duplicate of)
f01	ROUTE_CONTROL_GROUP	ROUTE_CONTROL_GROUP	
f02	ACD_WORK	ACD_WORK	
f03	ACD_LOGON	ACD_LOGON	
f04	ACD_LOGOFF	ACD_LOGOFF	
f05	ACD_AVAILABLE	ACD_AVAILABLE	
f06	ACD_UNAVAILABLE	ACD_UNAVAILABLE	
f07	SILENT_MONITORING	SILENT_MONITORING	
f08	TONE_MONITORING	TONE_MONITORING	

f09	CAF_LOGOFF	CAF_LOGOFF	
f10	CAF_GROUP_LOGOFF	CAF_GROUP_LOGOFF	
f11	CAF_AVAILABLE	CAF_AVAILABLE	
f12	CAF_UNAVAILABLE	CAF_UNAVAILABLE	
f13	CAF_GROUP_AVAILABLE	CAF_GROUP_AVAILABLE	
f14	CAF_GROUP_UNAVAILABLE	CAF_GROUP_UNAVAILABLE	
f15	CAF_FORWARD	CAF_FORWARD	
f16	CAF_FORWARD_CANCEL	CAF_FORWARD_CANCEL	
f17	ACD_PRIMARY_QUEUE	ACD_PRIMARY_QUEUE	
f18	ACD_PRIMARY_GROUP	ACD_PRIMARY_GROUP	
f19	ACD_SECONDARY_QUEUE	ACD_SECONDARY_QUEUE	
f20	ACD_SECONDARY_GROUP	ACD_SECONDARY_GROUP	
f21	ROLM_PARK	ROLM_PARK	
f22	TDS_MILLIWAT_TEST	TDS_MILLIWAT_TEST	
f23	TDS_LOOPBACK_TEST	TDS_LOOPBACK_TEST	
f24	TDS_SILENCE_TEST	TDS_SILENCE_TEST	
f25	TDS_COMBINATION_TEST	TDS_COMBINATION_TEST	
f26	SPKR_CALL_1_WAY	SPKR_CALL_1_WAY	
f27	SPKR_CALL_1_WAY_BCAST	SPKR_CALL_1_WAY_BCAST	
f28	HW_TEST_ON_SYMPH_ACT	HW_TEST_ON_SYMPH_ACT	
f29	EPNP_BARRIER_CODE	EPNP_BARRIER_CODE	
f30	EPNP_LOCATION_CODE	EPNP_LOCATION_CODE	
f31	EPNP_EVAL_EXTENSION	EPNP_EVAL_EXTENSION	
f32	API_WORKSTN_SESSION	API_WORKSTN_SESSION	
*	DATA_HUNT_GRP	DATA_HUNT_GRP	(duplicate of)
f33	CALL_ID_BLOCK	CALL_ID_BLOCK	
f34	CALL_ID_UNBLOCK	CALL_ID_UNBLOCK	

Object Type for Action Protocol

*	System	System	(duplicate of)
*	Station	Station	(duplicate of)
*	Call Pickup	Call Pickup	(duplicate of 604)

* f38	Hunt Group	Hunt Group	(duplicate of O55)
*	Subscriber	Subscriber	(duplicate of)
*	Extension	Extension	(duplicate of)
*	COS Changeover Group	COS Changeover Group	(duplicate of)
*	Class of Service	Class of Service	(duplicate of)
*	Speed Dial List	Speed Dial List	(duplicate of)
*	Standard Key Layout	Standard Key Layout	(duplicate of)
*	Individual Key Layout	Individual Key Layout	(duplicate of)
*	Alternate Voice Data	Alternate Voice Data	(duplicate of)
*	Associated DCI	Associated DCI	(duplicate of)
*	Multifunctional Terminal Adaptor	Multifunctional Terminal Adaptor	(duplicate of)
*	National ISDN	National ISDN	(duplicate of)
*	Stand alone DCI	Stand alone DCI	(duplicate of)
*	DCM DCO	DCM DCO	(duplicate of)
*	Stand alone DCM	Stand alone DCM	(duplicate of)
*	DCI Modem	DCI Modem	(duplicate of)
*	DCM Modem	DCM Modem	(duplicate of)
*	Code Conversion Device	Code Conversion Device	(duplicate of)
*	Fax	Fax	(duplicate of)
f40	Board	Baugruppe	
f41	Com Group	Com-Gruppe	
f42	Names	Namen	
f43	Hotline Index	Hotline-Index	
f44	Hunt Group Data	Daten Sammelanschluß	
f45	Hunt Group Fax	Fax Sammelanschluß	
*	ICOM	ICOM	(duplicate of)
f46	LCR Class of Service	LCR-Berechtigungsklasse	
f47	Port for General Answering	Anschluss für allg. Abfragen	
* f48	Night Options	Night Options	(duplicate of O78)
f49	PhoneMail COS	PhoneMail-COS	
f50	PhoneMail Mailbox	PhoneMail-Mailbox	
f51	PhoneMail System	PhoneMail-Anlage	

* f52	Home Location Codes	Home Location Codes	(duplicate of P39)
* f53	Dial Numbers	Dial Numbers	(duplicate of O93)
* f54	PIN	PIN	(duplicate of O86)
f55	System Speed Dial Lists	Zentralkurzwahlliste	
f56	Route Index	Richtungsindex	
f57	SNS Product System	SNS Product System	
f58	Teleworking Client	Teleworking Client	
f59	Xpressions470 Subscriber	Xpressions470 Subscriber	
f60	Xpressions470 COS	Xpressions470 COS	

Operations for Action Protocol

*	Add	Add	(duplicate of)
*	Delete	Delete	(duplicate of 632)
*	Modify	Modify	(duplicate of)
*	Information	Information	(duplicate of)
*	Single-phase Move	Single-phase Move	(duplicate of)
f64	Broadcast	Synchronisieren	

*	Adapt	Adapt	(duplicate of)
*	Save	Save	(duplicate of)
*	Upload All	Upload All	(duplicate of)
*	Upload Delta	Upload Delta	(duplicate of)
*	Remove Lock	Remove Lock	(duplicate of)

*	Include	Include	(duplicate of)

*	Verify	Verify	(duplicate of)
*	Add Name/PINs	Add Name/PINs	(duplicate of)
*	Delete Name/PINs	Delete Name/PINs	(duplicate of)
*	Modify Name/PINs	Modify Name/PINs	(duplicate of)
*	Add Member	Add Member	(duplicate of)
*	Delete Member	Delete Member	(duplicate of)
*	Modify Member	Modify Member	(duplicate of)

Operations for TW-Client

*	Basic	Basic	(duplicate of z40)
*	None	None	(duplicate of k54)
*	Standard	Standard	(duplicate of z41)
f67	Advanced	Fortgeschritten	

Operations for XPR470

f70	Subscribers can choose to be listed or not	Teilnehmer können wählen ob sie im Verzeichnis eingetragen werden wollen	
f71	Subscribers cannot change this setting for themselves	Teilnehmer können diese Einstellung nicht ändern	
*	None	None	(duplicate of k54)
f72	Exchange	Exchange	
f73	SMTP	SMTP	
f74	Lotus Notes	Lotus-Notes	
f75	Group Wise	Gruppenweise	

Exception Values

*	ILLEGAL	ILLEGAL	(duplicate of Z98)

**** subdomain codes continue later with code s00!!

10.3.73 coded values for stations (unity)

* dimsu's HDMS 6.0

* h05	API	API	* (duplicate of 244) number of optiset devices using workstation protocol
h06	ORG	ORG	* number of organizational unit

h07	COMGRP.	COMGRP.	* number of COM groups
* h08	KEY300	KEY300	* (duplicate of 557) number of KEY300 devices
h09	OptiPoint500	OptiPoint500	* number of VISION2000 devices

10.3.74 dynamic, opt_roomchar

* h15	ECHO	ECHO	* (duplicate of C25) echoing room
h16	NORMAL	NORMAL	* normal room
h17	QUIET	GDAEMPFT	* quiet room

** these 3 codes exist twice (see u75...)!!!!!!!!!!!!!!!!!!!!!!!!!

10.3.75 switch, upload_txt

h22	no upload is running and last upload was successful	Kein Upload aktiv; letzter Upload-Vorgang erfolgreich abgeschlossen	* upload = 0
h23	upload DELTA initiated (by Start update)	Upload DELTA eingeleitet (durch Update starten)	* upload = 1
h24	upload ALL initiated (by Start update)	Upload ALL eingeleitet (durch Update starten)	* upload = 2
h25	upload ALL running (initiated by User)	Upload ALL läuft (benutzerseitig eingeleitet)	* upload = 3
h26	upload DELTA running (initiated by User)	Upload DELTA läuft (eingeleitet durch Benutzer)	* upload = 4
h27	upload ALL reset by User	Upload ALL benutzerseitig zurückgesetzt	* upload = 5
h28	upload DELTA reset by User	Upload DELTA benutzerseitig zurückgesetzt	* upload = 6
h29	upload failed	Upload fehlgeschlagen	* upload = 7
h30	upload ALL necessary (e.g. version of the system has changed)	Upload ALL erforderlich (z.B.Version der Anlage wurde geändert)	* upload = 8

h31	upload ALL necessary (new system, has not been synchronized yet)	Upload ALL erforderlich (neuer Switch, noch nicht synchronisiert)	* upload = 9
h32	upload ALL for first synchronization initiated by User	Upload ALL für Erstsynchronisierung benutzerseitig eingeleitet	* upload = A
h33	upload has been postponed (by Start update)	Upload wurde verschoben (durch Update starten)	* upload = D

10.3.76 Devices

h42	OptiPoint 500	OptiPoint 500	* (MU9504)

10.3.77 port, offtype

h45	DC	MV	* deactivate with camp-on
h46	DI	OV	* deactivate immediatelly
h47	NL	NL	* no lock

10.3.78 port, pulslev

h90	VEFF3	VEFF3	* 3V r.m.s.
h91	VEFF5	VEFF5	* 5V r.m.s.

10.3.79 port, pulstype

h97	FORM	FORMAT	* pulse form reversal
h98	KHZ12	KHZ12	* 12 khz charge metering pulses
h99	KHZ16	KHZ16	* 16 khz charge metering pulses
i00	POL	POL	* static polarity reversal
i01	POL12	POL12	* static polarity reversal + 12 khz
i02	POL16	POL16	* static polarity reversal + 16 khz

10.3.80 port, ackst

i08	CHR	Geb	* start of announcement after first charge pulse
i09	TE	Te	* text end
i10	CHR&TE	GEB&TE	* previous two together

10.3.81 port, act

10.3.82 port, net_config

* i51	HTB	HTB	* (duplicate of 589) hicom trading board
* i52	TC	TC	* (duplicate of 590) telecommuting

10.3.83 port, cmind (berlm) and spec

i58	SUFDIAL	NAWA	* station can suffix dial
i59	SSTN	SOTL	* special station
i61	WITNS	ZEUGE	* stn/port for connect. as witness

10.3.84 port, ins (inbetr)

i67	ACL station	ACL-Teilnehmeranschluss	* acl support station
i68	ACL station logged in	ACL-Teilnehmeranschluss angemeldet	* acl support station

10.3.85 port, alloc

i73	ASCEND	AUFST	* ascending
i74	SAME	GLEICH	* same

10.3.86 port, sztype

i80	ATND	MELDEN	* seizure for evm attendant
i81	DSZCO	DRA	* direct seizure CO call
i82	DSZINT	DRI	* direct seizure internal
i83	EXPR	EBX	* express box seizure
* i84	FWD	FWD	* (duplicate of 853) call forwarding for hicom method
i85	FWDNANS	RWSN	* seizure after no answer forward

10.3.87 port, protocol_class

i86	TS	TS	
i87	WP	WP	
i88	WPPLUS	WPPLUS	

10.3.88 cfw, service

*cfw, service			
i90	VOICE	VOICE	* voice
i91	AUDIO3K1	AUDIO3K1	* modem 3,1 khz
i92	FAXG23	FAXG23	* fascimile group 2&3
i93	FAXG4	FAXG4	* fascimile group 4
i94	GENSRVC	GENSRVC	* general service
i95	SPR	SPR	* voice (DSS1)
i96	TEL3K1	TEL3K1	* telephony (3,1 khz, anate)
i97	TEL7K	TEL7K	* telephony-7khz
i98	TOENE	TOENE	* unrestricted digital information with tones
i99	UDI	UDI	* unrestricted digital information
j00	VIDEOTEL	VIDEOTEL	* video telephony

* j01	FAX	FAX	* (duplicate of 212)
* j02	DTE	DTE	* (duplicate of 219)

10.3.89 cfw, itype

*cfw, itype			
j03	EXTERN	EXTERN	* extern
j04	INTERN	INTERN	* intern
j05	GEN	GEN	* all calls

10.3.90 cfw, variant

*cfw, variant			
j07	STATION	STATION	* station
j08	SYSTEM	SYSTEM	* system

10.3.91 port, ring_type

j10	Audible Ring	Akustisches Rufsignal	* MU9512: Audible Ring

10.3.92 port, ops_conn_type

j12	Direct connection	Direkte Verbindung
j13	Off premise connection via T1 span	Externverbindung über T1-Strecke
j14	Off premise connection via SLMA	Externverbindung über SLMA

10.3.93 pin_check, typechkn

j16	COMPREM	KOMPREST	* complement of the remainder to modulus
j17	RESTDIV	RESTDIV	* rest of division

10.3.94 station_xref, xref_type

j20	Call Forward Dest.	Anrufuml.-Ziel	* Call Forward Destination
j21	Key Destination	Tastenziel	* Key Destination

10.3.95 pin_check, typesum

j23	CHKSUMPR	QRSUMPRO	* checksum of the products
j24	SUMPRO	SUMPRO	* sum of the products

10.3.96 pinclass, trkcdtyp

j30	PINCD	PINKZ	* autom. dial. of a trunk access code requires pin
j31	UNICD	ALGKZ	* autom. dial. univer. trk. access code follow. pin

10.3.97 pinclass, dispid

j38	PERMDISP	PERMANZ	* id is permanently displayed
j39	TIMEDISP	ZEITANZ	* id is displayed for 5 seconds

10.3.98 pinclass, cotype

j45	MOBILE	MOBIL	* mobile station
j46	BUSLONG	DSTLANG	* business call, long
j47	BUSSHRT	DSTKURZ	* business call, short
j48	NONE	NONE	* no identification
j49	PCODE	PKZ	* project code
j50	PRIVSHRT	PRIVKURZ	* private call, short

10.3.99 pinclass, transfer

j51	COS	COS	* will be not support from Call Processing
* j52	All	All	* (duplicate of C18)

10.3.100 pinclass, pindur

j57	ONE	Eins	* pin activated for 1 call
j58	PERM	Permanent	* pin is permanently active
* j59	TIME	TIME	* (duplicate of 883) pin is deactivated after timeout

10.3.101 pinclass, pintest

j65	MODULO1	Modulo 1	* modulo check algorithm 1
j66	MODULO2	Modulo 2	* modulo check algorithm 2
j67	MODULO3	Modulo 3	* modulo check algorithm 3
j68	TABLE	Tabelle	* check against pin tabelle
j69	LENGTH	Länge	

10.3.102 vbz, src??_dst??

j74	AA	EE	* allowed in both directions
j75	AD	EV	* allowed:source -> dest., denied: dest. -> source
j76	AX	EX	* allowed:source -> dest., dest.->source unchanged
j77	DA	VE	* denied:source -> dest., allowed:dest.-> source
j78	DD	VV	* denied in both directions
j79	DX	VX	* denied:source -> dest., other direction unchanged

10.3.103 vbz, card??_crdr??

j87	A	A	* allowed from card to cardreader
j88	D	D	* denied from card to cardreader

10.3.104 lcrauth, format

j94	ENCRYPT	KRYPT	* enter encrypted authorisation code
* j95	NORMAL	NORMAL	* (duplicate of h16) enter normal authorisation code

10.3.105 huntgrp, disptype

k01	EXT	EXT	* Display of ext. of conn. hunting group member
* k02	HT	HT	* (duplicate of 876) hunt group

10.3.106 key functions

k03	SYSSP1	SYSSP1	
k04	SYSSP2	SYSSP2	
k05	SPDI	SPDI	
k06	CDRACC	CDRACC	
k07	MENU	MENU	
k08	BUSYLAMP	BUSYLAMP	* MU9504: VISION 2000
k09	DOOROPEN	TORAUF	* MU9504: VISION 2000
k10	SPKRCALL	SPKRCALL	* MU9797: Intercom Feature
k11	BROADCST	BROADCST	* MU9797: Intercom Feature
k12	COMGRP	COMGRP	* MU9797: Intercom Feature

k13	COMSPK	COMSPK	* MU9797: Intercom Feature
k14	PREVIEW	PREVIEW	* MU9519: Preview Key
k15	PHML	PHML	* MU9533: Voice Mail
k16	RPTID	RPTID	* MU10064: Repeat ID
k17	CNCT	CNCT	
k18	BUZZ	BUZZ	
k19	SPARK	SPARK	
* k20	DCPA	DCPA	* (duplicate of n86)

10.3.107 berum, cosact

k22	COS1	COS1	* activate cos 1
k23	COS2	COS2	* activate cos 2

10.3.108 vbzgr, vbz_int??

k24	INT01	INT01	* internal zone
k25	INT02	INT02	* internal zone
k26	INT03	INT03	* internal zone
k27	INT04	INT04	* internal zone
k28	INT05	INT05	* internal zone
k29	INT06	INT06	* internal zone
k30	INT07	INT07	* internal zone
k31	INT08	INT08	* internal zone
k32	INT09	INT09	* internal zone

10.3.109 vbzgr, vbz_ank??

k40	INC01	ANK01	* incoming zone
k41	INC02	ANK02	* incoming zone
k42	INC03	ANK03	* incoming zone
k43	INC04	ANK04	* incoming zone

k44	INC05	ANK05	* incoming zone
k45	INC06	ANK06	* incoming zone
k46	INC07	ANK07	* incoming zone
k47	INC08	ANK08	* incoming zone
k48	INC09	ANK09	* incoming zone

10.3.110 night_opt, night_opt_type

k54	None	None	* NONE_NIGHT_OPT
k55	General Answering Station	Allgemeine Abfragestelle	* Night option for "general answer" (Nachtglocke)
* k56	Station	Station	* (duplicate of 601) Night option contains a list of up to 10 subscribers

* within night_opt_sub

k57	extern. centralized Atnd. Service Group	externe ZVGR-Variante	* CASEXT
k58	intern. centralized Atnd. Service Group	interne ZVGR-Variante	* CASINT
k59	Night Attendant	Nacht-Vermittlungsplatz	* NIGHT

10.3.111 spec_subscriber, spec_sub_type

k60	MSGR_	BOTE_	
k61	GENANS_	ABF_	* Allgem Abfragestelle
k62	CCS_	PSE	* Personensucheinrichtung
k63	VMXMWI	VMXMWI	* Common VoiceMail Interface
* k64	EXTANS	EXTANS	* (duplicate of n99) Externes Ansagegeraet

10.3.112 port, speed_dial

k71	max. 10 entries	max. 10 Einträge
k72	max. 20 entries	max. 20 Einträge
k73	max. 30 entries	max. 30 Einträge

10.3.113 schedule batch, sb_state_txt

k77	Scheduled batch opened	Zeitauftrag geöffnet	* sb_open state=o
k78	Execution time deleted	Ausführzeit gelöscht	* sb_unsched state=u
k79	Execution time set	Ausführzeit gesetzt	* sb_sched state=s
k80	Scheduled batch verification	Zeitauftrag-Prüfung	* sb_verifystate=v
k81	Scheduled batch closed	Zeitauftrag geschlossen	* sb_close state=c
k82	Open for verification	Offen für Prüfung	* sb_verifyopen state=r
k83	Scheduled batch failed	Zeitauftrag fehlgeschlagen	* sb_failed state=f
k84	Execution of Scheduled batch	Ausführen von Zeitauftrag	* sb_execute state=e

10.3.114 Coded values for dialplan

priv_dial_number, type

* n05	STN	STN	* (duplicate of 102) station
n06	DATA_DEST	DATA_DEST	*
n08	DATA_HUNT_GRP	DATA_HUNT_GRP	*

route index, type

n09	DPLN	WABE	* DPLN route
* n10	LCR	LCR	* LCR route (duplicate of C16)

route index, route_preference

n11	primary	Primär	* Primary route
n12	alternate	alternativ	* Alternate route

route index, connection

n13	direct	direkt	* Direct connection
n14	indirect	indirekt	* Indirect connection

dial_code, code_type

n15	GENANS	ABF	* General answering
n16	SIGNOFF	SIGNOFF	*
* n17	ACC	ACC	* (duplicate of 936) Account Code
* n18	ACDWORK	ACDWORK	* (duplicate of 922) Agent work
n19	ACDSHMSG	ACDKURZN	* Short message
n20	ACDLOGOF	ACDLOGOF	* Agent logoff
n21	ACDLOGON	ACDLOGON	* Agent logon
* n22	ACDEMMSG	ACDEMMSG	* (duplicate of 930) Emergency message)
* n23	ACDNAV	ACDNAV	* (duplicate of 923) Agent not available
* n24	ACDPGS	ACDPGS	* (duplicate of 924) Primary group status
* n25	ACDPQS	ACDPQS	* (duplicate of 926) Primary queue status
* n26	ACDSGS	ACDSGS	* (duplicate of 925) Secondary group status
* n27	ACDSQS	ACDSQS	* (duplicate of 927) Secondary queue status
* n28	ACDAV	ACDAV	* (duplicate of 921) Agent available
* n29	CO	CO	* (duplicate of 104) Central office
n30	KNOVRKY	ANKASTA	* Knocking - Override - key
* n31	KNOVR	KNOVR	* (duplicate of 851) Knocking - Override
n32	SIGNON	ANMELDEN	* Sign on a line via terminal device
n33	ANSCO	ANSEAMT	* Answering device for central office
n34	ANSINT	ANSEHAUS	* Answering device - internal
n35	DNDKY	ANSTA	* Do not disturb
n36	IUSEKY	ANZLEITA	* Show-line-ley-I-use-key
n37	DISUOFF	ANZUAUS	* Display suppression off
n38	DISUON	ANZUEIN	* Display suppression on

n39	DDND	ARSTA	* Deactivate do not disturb
n40	ADND	ARSTE	* Activate do not disturb
n41	OVRPRTCT	OVRPRTCT	* Override protection activation
n42	ASTN	ATLN	* Check A-station number in case of transit
n43	DFWDVTX	AULABTX	* Deactivate call forwarding VTX
n44	DFWDDTE	AULADEE	* Deactivate call forwarding DTE
n45	DFWDFAX	AULAFAX	* Deactivate call forwarding FAX
n46	DFWDREM	AULAREM	* Deact call forwarding remote - follow me
n47	DFWDVCE	AULASPR	* Deactivate call forwarding VOICE and TTX
n48	AFWDVTX	AULEBTX	* Activate call forwarding VTX
n49	AFWDDTE	AULEDEE	* Activate call forwarding DTE
n50	AFWDDWD	AULEDOD	* Activate call forwarding for DIGITE without dsiplay
n51	AFWDFAX	AULEFAX	* Activate call forwarding FAX
n52	FWDTERM	AULEG	* Call forwarding for all stns assoc. to terminal
n53	AFWDREM	AULEREM	* Act call forwarding remote - follow me
n54	AFWDVCE	AULESPR	* Activate call forwarding VOICE and TTX
n55	DFFWDVTX	AULFABTX	* Deactivate fixed forwarding VTX
n56	DFFWDDTE	AULFADEE	* Deactivate fixed forwarding DTE
n57	DFFWDFAX	AULFAFAX	* Deactivate fixed forwarding FAX
n58	DFFWDVCE	AULFASPR	* Deactivate fixed forwarding VOICE
n59	AFFWDVTX	AULFEBTX	* Activate fixed forwarding VTX
n60	AFFWDDTE	AULFEDEE	* Activate fixed forwarding DTE
n61	AFFWDFAX	AULFEFAX	* Activate fixed forwarding FAX
n62	AFFWDVCE	AULFESPR	* Activate fixed forwarding VOICE
n63	FWDIGNOR	AULIGNOR	* Ignore call forwarding

n64	FWDREM	AULREM	* Show call forwarding remote-follow me
n65	FWDKY	AULTA	* Call forwarding key
* n66	PU	PU	* (duplicate of 854) Call pickup
n67	PUGDIS	AUNGANZ	* Display group pickup information on request
n68	PUDIR	AUNGEZ	* Directed call pickup
n69	BABYLSNG	BABYUEBW	* Baby listening
n70	DCOSX	BERUMA	* Deactivate COS changeover from station
n71	ACOSX	BERUME	* Activate COS changeover from station
n72	MSGRKY	BORUTA	* Messenger key
n73	MBKY	BRIKATA	* Mailbox key
n74	VTXNO	BTXRUNU	* Combi-code for videotex terminal
n75	CAFDFWD	CAFAULA	* CAF deactivate forwarding of RCG
n76	CAFAFWD	CAFAULE	* CAF activate forwarding of RCG
n77	CAFGRNAV	CAFGRNVB	* CAF group not available
n78	CAFGROFF	CAFGROFF	* CAF group logoff
n79	CAFGRAV	CAFGRVB	* CAF group available
n80	CAFLOGOF	CAFLOGOF	* CAF agent logoff
n81	CAFNAV	CAFNVB	* CAF agent not available
n82	CAFAV	CAFVB	* CAF agent available
n83	CALLPARK	CALLPARK	* Call park for Anate
n84	OPROVR	CALLTOCO	* Demand to override by co operator
n85	DAKY	DATA	* Non-voice (data) key
n86	DCPA	DCPA	* Park a user at a specific user
n87	DTEDNO	DEEDRUNU	* Combi-code for DTE terminal (digital traffic)
n88	DTENO	DEERUNU	* Combi-code for DTE terminal (analog traffic)
n89	DIGIDAT	DIGIDAT	* Digite data input

n90	DICT	DIBE	* Dictation equipement
n91	VOICECAL	DIRANS	* Voice calling
n92	VCRKY	DIRSTA	* Voice call reject key
n93	VCKY	DIRTA	* Voice call key
n94	DSSKY	DIRUTA	* Direct station select key
n95	CONF3	KF3	* Threeway conference
n96	RA	DUSABE	* Recorded announcement
n97	OWNNODE	EIGENKZ	* Code for own node
n98	EXHOLDKY	EXHALTTA	* Exclusive hold key
n99	EXTANN	EXTANS	* Interception by console in partner system
o00	TRACE	FGAUT	* Malicious call tracing
o01	FAXNO	FAXRUNU	* Combi-code for fax terminal
o02	FLTRACE	FLTRACE	* Activate flag trace
o03	FREQTEST	FREQTEST	* DTMF - MFC transmitter test
o04	CTLS	GLTA	* Controlled trunk and line selection
o05	HOLDKY	HALTETA	* Hold key
o06	HWTEST	HWTEST	* Activate HW test on terminal device (Optiset)
o07	KEYPADIE	KEYPADIE	* Keypad IE
o08	CONFKY	KONFTA	* Conference key
* o09	SPDI	SPDI	* (duplicate of k05) DB_KZP_SP calling individual
o10	SPDIPROG	KUWAIPRO	* DB_KZP_SP call individual programming code
o11	SPDC1	RNGZ1	* DB_KZP_SP calling central 1
o12	SPDC2	RNGZ2	* DB_KZP_SP calling central 2
* o13	VTX	VTX	* (duplicate of 213) Code for VTX service
* o14	DTE	DTE	* (duplicate of 219) Code for DTE service (analog traffic)
* o15	DTED	DTED	* (duplicate of 217) Code for DTE service (digital traffic)

* o16	FAX	FAX	* (duplicate of 212) Code for FAX service
o17	SPLIT	SPLIT	* Split - alternating calls
o18	MANDOWNL	MANDOWN	* Manual downloading
o19	TOGGLEKY	MATA	* Toggle key
* o20	ATND	ATND	* (duplicate of i80) Attendant - internal operator code
o21	DTMFCONV	MFVUMS	* DTMF-conversion when dialing to public network
o22	MONSLNT	MITHOER	* Silent monitoring
* o23	MONTONE	MONTONE	* (duplicate of 934) Tone monitoring
o24	MBOFF	MWIAUS	* Mailbox LED off
o25	MBON	MWIEIN	* MAilbox LED on
o26	NAMEKY	NATA	* Name key
o27	NAKYLO	NATALO	* Allocate local name key (for digite-M)
o28	NOPT	NAVAR	* Change of night switch option
o29	NETW	NETZ	* DAR for a voice feature in another node
o30	NETWDTE	NETZDEE	* DAR for a DTE feature in another node
o31	NETWFAX	NETZFAX	* DAR for a FAX feature in another node
o32	NETRTE	NETZRTG	* Auxiliary DAR for networking
o33	EOVR	NATR	* Emergency override
o34	PARKKY	PARKTA	* Park key
o35	DPIN	PINAUS	* Deactivate PIN
o36	APIN1	PINE1	* Activate PIN for PIN1
o37	APIN2	PINE2	* Activate PIN for PIN2
o38	APIN3	PINE3	* Activate PIN for PIN3
o39	APIN4	PINE4	* Activate PIN for PIN4
o40	APIN5	PINE5	* Activate PIN for PIN5
o41	DPRIV	PRIVA	* Deactivate privacy
o42	APRIV	PRIVE	* Activate privacy

o43	PRIVKY	PRIVTA	* Privacy key
o44	TESTLN	PRUEFANS	* Test line
o45	CCDIS	PSEKODI	* Code calling display with code
o46	CCMEETME	PSEKOM	* Code calling meet me with code
o47	CCSTN	PSEKORU	* Code calling station A with code
o48	CCVCE	PSEKOSPR	* Code calling for voice with code
o49	CCSURG	PSEKOSUD	* Code calling urgent search with code
o50	CCS	PSEKOSUN	* Code calling normal search with code
o51	CCANS	PSEMEL	* Code calling answer
o52	CCSCD	PSESUKO	* Code calling search with code
o53	CCSN	PSESUN	* Code calling search normal
o54	CCMANS	PSMMEL	* Code calling answer to mult. task page attempt
o55	CCMSURG	PSMSUD	* Code calling multiple task paging urgent
o56	CCMS	PSMSUN	* Code calling multiple task paging normal
o57	CCKY	PSTA	* Code calling key
* o58	TIE	TIE	* (duplicate of 105) Tie line access
* o59	TIEATND	TIEATND	* (duplicate of 106) Tie trunk access-attendant console
o60	RCG	RCG	* Route control group number
o61	RECONNPA	RECONNPA	* Reconnect on held call
o62	RERING	RERING	* Ring partner again
* o63	SCONSCO	SCONSCO	* (duplicate of 107) Sat. connection in consultation hold from CO
* o64	SCONSI	SCONSI	* (duplicate of 108) Sat. connection in consultation hold from internal
o65	CONSKY	RFTA	* consultation key
o66	CBKKY	RRTA	* Call back key

o67	DCBK	RRVOLA	* Delete callback
o68	ACBK	RRVOLE	* Activate callback
o69	RESDOWNL	RSTDOWN	* Rest by downloading
o70	RCTOFFKY	RUFAUSTA	* Ringet cutoff key
o71	DHTVTX	SAABTX	* Disassociate station from hunt group, VTX eqpt.
o72	DHTDTE	SAADEE	* Disassociate station from hunt group, DTE eqpt.
o73	DHTFAX	SAAFAX	* Disassociate station from hunt group, FAX eqpt.
o74	DHTVCE	SAASPR	* Disassociate station from hunt group, voice-TTX eqpt.
o75	AHTVTX	SAEBTX	* Add station to hunt group VTX
o76	AHTDTE	SAEDEE	* Add station to hunt group DTE
o77	AHTFAX	SAEFAX	* Add station to hunt group FAX
o78	AHTVCE	SAESPR	* Add station to hunt group voice and TTX
* o79	HTG	HTG	* (duplicate of 101) Hunting group
o80	HTKY	SATA	* Hunt group key
o81	SELFPA	SELFPA	* Park a user on one's own line
o82	FLASHCD	SIGTA	* Equivalent flash signal code
o83	SPD	SPEED	* Changing speed
o84	NOEC	NOEC	* CLC (Clear-Channel) without echo cancellation for encrypted calls

* o89	CTDL	CTDL	* (duplicate of 110) Cut-through or suffix dialing in tie traffic
o90	STKY	STATA	* Start key
o91	KYPROG	TAPRO	* Key programming on Digite
o92	DREM	TERMA	* Delete reminder
o93	AREM	TERME	* Activate reminder
o94	REMKY	TERMINTA	* Reminder key
o95	TESTTONE	TESTTONE	* Connect test ton transmitter for 800_Hz test

* o96	STN	STN	* (duplicate of 102) Station
o97	STNREL	TLNTR	* Stn DAR for destination-dependent release
o98	DOOROPEN.	TORAUF.	* Door open
o99	DOORCL	TORGESPR	* Door call
p00	RELEASE	TRENNEN	* Emergency release of a station
p01	RLSKY	TRTA	* Release-cancel key
p02	PUEXT	UEBEXT	* Acceptance (or pickup) of ext. consult call
p03	TRNSKY	UEGTA	* Call transfer key
p04	TIMEKY	UHRTA	* Date-time key
* p05	SAT	SAT	* (duplicate of 109) Extend indial or trunk call to satellite system
p06	VMOWN	VMEIGEN	* Access to own voice mailbox
p07	VMFOR	VMFREMD	* Access to another ('foreign') user's voice mailbox
* p08	AC Call Number	AC Call Number	* (duplicate of H20) Individual attentant console
* p09	Attendant Console	Attendant Console	* (duplicate of 103) Indial to attendant console
* p10	SNR	SNR	* (duplicate of 885) Saved number redial
* p11	LNR	LNR	* (duplicate of 504) Last number redial
p12	LNRKY	WAWILTA	* Last number redial key
p13	SNRKY	WAWITA	* Saved number redial key
p14	WSS	WSS	* Establish a workstation session
* p15	OUTPULSE	OUTPULSE	* (duplicate of C31) Outpulse specific digits
p16	NWS	NWS	* night watchman service

dial_code, check

p21	ALL	ALLE	* All
p22	N	N	* No
p23	PART	TEIL	* Partial check during add action

dial_code, cfremvar

* p29	CFB	CFB	* (duplicate of 842) Call forwarding - busy
* p30	CFNR	CFNR	* (duplicate of 843) Call forwarding - no reply
* p31	CFU	CFU	* (duplicate of 841) Call forwarding - unconditional

dial_code, cfremse

* p37	ALL	ALL	* (duplicate of p21) All
* p38	AUDIO3K1	AUDIO3K1	* (duplicate of i91) Audio 3,1 kHz (Modem)
p39	VTXH	BTXH	* Videotex Syntax based (HICOM VTX)
p40	DTEH	DEEH	* DTE (Hicom)
p41	FAX3	FAX3	* Fax group 2/3
p42	FAX4	FAX4	* Fax group 4
p43	FAXH	FAXH	* Fax group 2/3 and 4 (HICOM FAX)
p44	SPEECH	SPRACHE	* Voice
* p45	TEL3K1	TEL3K1	* (duplicate of i96) Telephony 3,1 kHz (Anate)
* p46	TEL7K	TEL7K	* (duplicate of i97) Telephony 7 kHz
p47	TTXH	TTXH	* Teletex (HICOM TTX)
* p48	UDI	UDI	* (duplicate of i99) Unrestricted digital Information
p49	UDITA	UDITA	* UDI with tones and announcements
p50	VIDEO	VIDEO	* Videotelephony
* p51	VOICE	VOICE	* (duplicate of i90) Voice (Hicom)

knmat, npi

p57	ISDN	ISDN	* ISDN-E.164 numbering plan
p58	PRIVATE	PRIVATE	* Private numbering plan
p59	UNKNOWN	UNKNOWN	* implicit numbering plan

knmat, onnol

p65	ROW	ALLKN	* Rest of world
p66	OWN	EIGEN	* Own node
p67	CDR	GE	* Charge data recording

knmat, conn

* p73	A	A	* (duplicate of j87) Allowed
* p74	D	D	* (duplicate of j88) Denied

numbering_plan, partner

* p77	OWN	OWN	* (duplicate of p66)
p78	FOREIGN	FREMD	

knpre, ton

p80	INTERNAT	INTERNAT	* International number
p81	LEVEL1	LEVEL1	* Level1-regional number
p82	LEVEL2	LEVEL2	* Level2-regional number
* p83	LOCAL	LOCAL	* (duplicate of I01) Local number
p84	NATIONAL	NATIONAL	* National number
p85	SUBSCR	SUBSCR	* Subscriber

knfor, scre

p91	NETWPROV	NETWPROV	* Network provided number
p92	TRANSP	TRANSP	* Transparent screening
p93	USERNSCR	USERNSCR	* Number not user screened
p94	USERVF	USERVF	* Number verification by user failed
p95	USERVP	USERVP	* Number verification by user passed

knfor, erropt

p97	DOMIN	DOMIN	* Configured tonout dominates in error case
p98	IGNOR	IGNOR	* Configured tonout ignored in error case

p99	IMPL	IMPL	* Tonout is implicit station no in error case

knfor, sendcall

q01	SEND0	SEND0	* Send no number
q02	SEND1	SEND1	* Send one number
q03	SEND2	SEND2	* Send two numbers

knfor, npi

q05	INTERPRE	INTERPRE	
q06	NATIOPRE	NATIOPRE	

rufum, type

q09	DID	DUWA	* Direct in dial
q10	IDN	IDN	* Integrated data network
q11	COMB	KOMBI	* Combination code
q12	COMB_FAX	COMB_FAX	* Combination code FAX
q13	COMB_BTX	COMB_BTX	* Combination code BTX
q14	COMB_DEE	COMB_DEE	* Combination code DEE
q15	STNO2	RUNU2	* Dest station number

lcr, lcr_attr

q21	OFFHOOKQ	OFFHOOKQ	*
q22	REROUTE_INTERNAL	REROUTE_INTERNAL	* No routing

priv_dial_number_ranges, dms_typer

q28	CMI	CMI	* Cordless
q29	EXT_CHESE	EXT_CHESE	* External chese

lcr, cbmattr

q35	NOCBM	NOCBM	* No byte manipulation
q36	EC64	EC64	* Only echo cancellation 64 msec
q37	EC64AMC	EC64AMC	* Echo cancellation 64 msec with A-u conversion
q38	EC128	EC128	* Only echo cancellation 128 msec

q39	EC128AMC	EC128AMC	* Echo cancellation 128 msec with A-u conversion
q40	AMC	AMC	* Only A-u conversion

miscellaneous

q45	COPIN10	COPIN10	* Activate PIN for copin10

dial_code,code_type again (dar/kzp)

q46	APIN11	PINE11	* activate pin for copin11
q47	APIN12	PINE12	* activate pin for copin12
q48	APIN13	PINE13	* activate pin for copin13
q49	APIN14	PINE14	* activate pin for copin14
q50	APIN15	PINE15	* activate pin for copin15
q55	PROT	PROT	* MLPP protected call
q56	PRTY	PRTY	* MLPP priority call - low level
q57	IMMED	IMMED	* MLPP immediate call
q58	FLASH	FLASH	* MLPP flash call
q59	FLASHOV	FLASHOV	* MLPP flash override call - high level
q60	FRCALTRT	ALTROUTE	* force alternate route
q61	INCALTRT	ALTRTKOM	* incoming alternate route
q62	DFWDEXT	AULAEXT	* deactivate call forwarding extern
q63	DFWDINT	AULAINT	* deactivate call forwarding intern
q64	AFWDB	AULEB	* activate call forwarding busy
q65	AFWDBNA	AULEBKA	* activate call forwarding busy or no answer
q66	AFWDEXIN	AULEEXIN	* activate call forwarding ext and int
q67	AFWDEXT	AULEEXT	* activate call forwarding extern
q68	AFWDINT	AULEINT	* activate call forwarding intern
q69	AFWDNA	AULEKA	* activate call forwarding no answer

q70	FFWDREMK	AULFREMT	* store call forwarding via remote by key
q71	DSYSFWD	AULSYSA	* deactivate system call forwarding voice
q72	ASYSFWD	AULSYSE	* activate system call forwarding voice
q73	BROADCST.	BROADCST.	* speaker call - one way broadcast
* q74	BUZZ	BUZZ	* alert (buzz) a target station (duplicate of k19)
* q75	CDRACC	CDRACC	* cdr account code (duplicate of k06)
q76	COMBO	COMBO	* combo -- transmission test termination
* q77	COMGRP	COMGRP	* com group call (duplicate of h07)
q78	COMSPK.	COMSPK.	* com group speaker call - two way
* q79	CONF	CONF	* establish conference / add party to conference (duplicate of 864)
* q80	HOLD	HOLD	* call hold by an analog phone (duplicate of 891)
q81	LOOPBACK	LOOPBACK	* loopback -- transmission test termination
q82	MILLWAT	MILLWAT	* milliwatt -- transmission test termination
q83	MWACT	MWACT	* message waiting activation
q84	MWANS	MWANS	* message waiting answer
q85	MWCAN	MWCAN	* message waiting cancel by terminating party
q86	MWCANORI	MWCANORI	* message waiting cancel by originating party
* q87	PARK	PARK	* call park and retrieve to system slot (duplicate of 871)
q88	APIN10	PINE10	* activate pin for copin10
q89	PRITEST	PRITEST	* primary rate test line loopback
q90	RMCONFLP	RMCONFLP	* remove last party from conference

q91	RMCONFP1	RMCONFP1	* remove party at position 1 from conference
q92	RMCONFP2	RMCONFP2	* remove party at position 2 from conference
q93	RMCONFP3	RMCONFP3	* remove party at position 3 from conference
q94	RMCONFP4	RMCONFP4	* remove party at position 4 from conference
q95	RMCONFP5	RMCONFP5	* remove party at position 5 from conference
q96	RMCONFP6	RMCONFP6	* remove party at position 6 from conference
q97	RMCONFP7	RMCONFP7	* remove party at position 7 from conference
q98	RMCONFP8	RMCONFP8	* remove party at position 8 from conference
q99	SILENCE	SILENCE	* silence -- transmission test termination
r00	SPKRCALL.	SPKRCALL.	* speaker call - one way
r01	CDFAX	KZFAX	
r02	CDDTE	KZDEE	
r03	DSNR		

lcr, svcvce

r05	FX	FZP	* Foreign Exchange Access
r06	LDS	LDS	* Long distance service
* r07	NONE	NONE	* No attributes (duplicate of 030)
r08	OUTWATS	OUTWATS	* Out wide area telecommunication service
r09	TTA	TTA	* Tie trunk access
r10	VPN	VPN	* Virtual private network
r11	WB0	WB0	* WATS Band 0
r12	WB1	WB1	* WATS Band 1
r13	WB2	WB2	* WATS Band 2
r14	WB3	WB3	* WATS Band 3
r15	WB4	WB4	* WATS Band 4

r16	WB5	WB5	* WATS Band 5
r17	WB6	WB6	* WATS Band 6
r18	WB7	WB7	* WATS Band 7
r19	WB8	WB8	* WATS Band 8
r20	WB9	WB9	* WATS Band 9

lcr, svcnvce

* r24	FX		* Foreign Exchange access (duplicate of r05)
* r25	NONE		* No attributes (duplicate of 030)
* r26	OUTWATS		* Out wide area telecommincation service (duplicate of r08)
r27	SDS		* Switched digital service
* r28	TTA	TTA	* Tie trunk access (duplicate of r09)
* r29	VPN	VPN	* Virtual private network (duplicate of r10)

lcrrule_sub, rule_element again

r34	SENDCIC	SENDCIC	* Send carrier identification code
r35	NWOP	NWOP	* Network operator
r36	CPNLIN	CPNLIN	* Send an up 22 to digit CPN or LIN
r37	SENDKEYP	SENDKEYP	* Dial info will be sent as keypad info element

lcrrule_sub, param

r40	ALTOP	ALTOP	* Request alternate operator
r41	PRINOP	PRINOP	* Request principal operator

port, ton

r45	ABBR	ABBR	* abbreviated
* r46	INTERNAT	INTERNAT	* international (duplicate of p80)
r47	L1REGIO	L1REGIO	* level 1 regional

r48	L2REGIO	L2REGIO	* level 2 regional
* r49	LOCAL	LOCAL	* local (duplicate of l01)
* r50	NATIONAL	NATIONAL	* national (duplicate of p84)
* r51	SUBSCR	SUBSCR	* subscriber (duplicate of p85)
* r52	UNKNOWN	UNKNOWN	* unknown (duplicate of p59)

port, npi

* r55	ISDN	ISDN	* isdn (e.164) (duplicate of p57)
* r56	PNP	PNP	* private (duplicate of p58)
* r57	UNKOWN	UNKOWN	* unknown (duplicate of p59)

numbering_plan,isdn_ul

* r60	EXT	EXT	* (duplicate of k01)
r61	INT	INT	
r62	NAT	NAT	
r63	SUB	TLN.	

numbering_plan,pnp_ul

* r67	EXT	EXT	* (duplicate of k01)
r68	L0	L0	
r69	L1	L1	
r70	L2	L2	

10.3.115 Codes for subdomain server

General Values

*	no	no	(duplicate of 0)
*	yes	yes	(duplicate of 1)
*	free	free	(duplicate of 2)

*	used	used	(duplicate of 3)

*	german	german	(duplicate of)
*	english	english	(duplicate of)

Switch Types

s00	ROLM9006	ROLM9006	

Switch Versions

s05	V6.5	V6.5	
s06	V6.6	V6.6	

Upload Values

*	Daily	Daily	(duplicate of)
*	Monday	Monday	(duplicate of)
*	Tuesday	Tuesday	(duplicate of)
*	Wednesday	Wednesday	(duplicate of)
*	Thursday	Thursday	(duplicate of)
*	Friday	Friday	(duplicate of)
*	Saturday	Saturday	(duplicate of)
*	Sunday	Sunday	(duplicate of)

SessionLog Values

*	Done without error	Done without error	(duplicate of)
*	Done with error	Done with error	(duplicate of)
*	No response	No response	(duplicate of)
*	DMS Error	DMS Error	(duplicate of)
*	Running	Running	(duplicate of)
*	No Error	No Error	(duplicate of)
*	Error	Error	(duplicate of)
*	Done	Done	(duplicate of)

Voice Device Types

s10	Analog	Analog
s11	Analog TA	Analog TA
s12	DYAD	DYAD
s13	DYADJr	DYADJr
s14	Off-Premise	Extern
s15	Off-Premise T1	Off-Premise T1
s16	Optiset E	Optiset E
s17	RP120	RP120
s18	RP240	RP240
s19	RP312	RP312
s20	RP400	RP400
s21	RP612	RP612
s22	RP624	RP624
s23	Phantom	Phantom
s24	Rolm Phone	Rolm Phone

Data Device Types

s30	Alternate Voice Data	Sprache/Daten-Wechsel
s31	Associated DCI	Assoziierte DCI
s32	Code Conversion Device	Codeumsetzungsgerät
s33	DCI Modem	DCI-Modem
s34	DCM DCO	DCM DCO
s35	DCM Modem	DCM-Modem
s36	Multifunctional Terminal Adaptor	Multifunktions-Endgeräteadapter
s37	National ISDN	National ISDN
s38	Stand alone DCI	Eigenständige DCI
s39	Stand alone DCM	Eigenständiges DCM

Layout Types

*	Analog	Analog	(duplicate of s10)
*	Analog TA	Analog TA	(duplicate of)

*	DYAD	DYAD	(duplicate of)
*	DYADJr	DYADJr	(duplicate of)
*	Off-Premise	Off-Premise	(duplicate of s15)
*	Off-Premise T1	Off-Premise T1	(duplicate of s14)
s43	OptiE - 12	OptiE - 12	
s44	OptiE - 3	OptiE - 3	
s45	OptiE - 8	OptiE - 8	
*	RP120	RP120	(duplicate of)
*	RP240	RP240	(duplicate of)
*	RP312	RP312	(duplicate of)
*	RP400	RP400	(duplicate of)
*	RP612	RP612	(duplicate of)
s46	RP612K	RP612K	
*	RP624	RP624	(duplicate of)
s47	RP624K	RP624K	
s48	OPTIKEYM	OPTIKEYM	
*	NI1200	NI1200	(duplicate of)

Values for Phonemail

s50	US English	US English	
*	French	French	(duplicate of C81)
*	Japanese	Japanese	(duplicate of 532)
*	UK English	UK English	(duplicate of 533)
*	German	German	(duplicate of 6)
s51	TDD US English	TDD US English	
s52	Default Language	Standard Sprache	
s53	PM6.1	PM6.1	
s54	PM6.2	PM6.2	
s55	PM6.3	PM6.3	
s56	PM6.3.3	PM6.3.3	

s57	PM6.4.3	PM6.4.3
s58	1.0	1.0
s59	37622	37622
s60	37653	37653
s61	2.0	2.0
s62	37623	37623
s63	37654	37654
s64	37743	37743
s65	3.0	3.0
s66	4.0	4.0

Values for Table remote

s70	standard	standard	
*	alternate	alternate	(duplicate of n12)

s72	LAN	LAN	
s73	V24	V24	
*	ISDN	ISDN	(duplicate of p57)
s75	MSV	MSV	

s76	1200	1200
s77	2400	2400
s78	4800	4800
s79	9600	9600
s80	14400	14400
s81	19200	19200
s82	28800	28800
s83	38400	38400
s84	56000	56000
s85	57600	57600

*	None	None	(duplicate of k54)
s87	Even	Gerade	
s88	Odd	Ungerade	

* s89	7	7	(duplicate of 054)
* s90	8	8	(duplicate of 055)

s91	1	1
s92	1.5	1.5
s93	2	2

s94	IN	IN
s95	OUT	OUT
s96	INOUT	INOUT

s97	On	An
s98	Off	Aus

s99	RMX	RMX
t00	UNIX	UNIX

t01	US Robotics Sportster FLASH x2	US Robotics Sportster FLASH x2
t02	3Com US Robotics 56K Message Modem	3Com US Robotics 56K Message Modem
t03	ELSA Multimodem V.34plus	ELSA Multimodem V.34plus
t04	3Com US Robotics 56k Voice Pro	3Com US Robotics 56k Voice Pro
t05	3Com US Robotics 56k Fax	3Com US Robotics 56k Fax
t06	MultiTech MultiModem ZDX 33.6k	MultiTech MultiModem ZDX 33.6k

t07	ONLINE	ONLINE
t08	OFFLINE	OFFLINE
t09	Enabled	Aktiviert
t10	Disabled	Deaktiviert

Values for EPNP

*	primary	primary	(duplicate of n11)
*	alternate	alternate	(duplicate of n12)

t15	NBC	NBC
t16	IBC	IBC

*next 2 codes new for DMS60:

t17	conflict with TWABE	Im Konflikt mit TWABE
t18	conflict with dial code	Im Konflikt mit Wählkennzahl

t19	no_loc_routing	no_loc_routing
t20	wrong_loc_routing	wrong_loc_routing
t21	indirect_loc_routing	indirect_loc_routing
t22	direct_loc_routing	direct_loc_routing
t23	no_net_routing	no_net_routing
t24	wrong_net_routing	wrong_net_routing
t25	indirect_net_routing	indirect_net_routing
t26	direct_net_routing	direct_net_routing
t27	conflict_loc_routing	conflict_loc_routing
t28	multiple_number	multiple_number

*	direct	direct	(duplicate of n13)
*	indirect	indirect	(duplicate of n14)

Values for LCT-API_ADP_Unit

t30	LBC	LBC
t31	LCX	LCX
t32	LCX_CHAN	LCX_CHAN
t33	IO	IO
t34	IO_DEV	IO_DEV
t35	IO_AREA	IO_AREA

Values for LCT-API_ADP_UnitType

t40	NO_UNIT_TYPE	NO_UNIT_TYPE	
*	LBC	LBC	(duplicate of t30)
t42	LBC_M	LBC_M	
t43	LCU	LCU	
t44	LCU_M	LCU_M	
*	LCR	LCR	(duplicate of C16)
t46	LCC	LCC	
t47	VCD	VCD	
t48	RAX	RAX	
t49	FCD	FCD	
t50	DP	IWV	
t51	DP_286	DP_286	
* t52	IP	IP	(duplicate of 267)
t53	MAC	MAC	
t54	MEM512	MEM512	
t55	MEM2M	MEM2M	
t56	IOPD	IOPD	
t57	IOPS	IOPS	
t58	D_DC	D_DC	
t59	C_DC	C_DC	
*	DC	DC	(duplicate of h45)
t61	FD	FD	
t62	HD	HD	
t63	CT	CT	
*	MB	MB	(duplicate of 856)
t65	MULTIBUS	MULTIBUS	
t66	IEC_625	IEC_625	
t67	LBU	LBU	
t68	MIP	MIP	
t69	IOPA	IOPA	
t70	LCS0_D	LCS0_D	

t71	PSIO	PSIO
t72	DMP	DMP
t73	DM80E	DM80E
t74	HUBC	HUBC
t75	DM3L	DM3L
t76	DM4L	DM4L
t77	DSC80	DSC80

Values for LCT-API_AlarmLevel

t80	SERVICE	SERVICE
t81	MINOR	MINOR
t82	MAJOR	MAJOR

Values for LCT-API_Analog_TA_Type

t85	ANALOG_TA_AB	ANALOG_TA_AB
t86	ANALOG_TA_AI	ANALOG_TA_AI
t87	ANALOG_TA_ON_NI_SYM	ANALOG_TA_ON_NI_SYM
t88	ANALOG_TA_ON_NI_NI1200	ANALOG_TA_ON_NI_NI1200

Values for LCT-API_AnswerMode

t90	AUTO	AUTO
t91	MANUAL	MANUAL

Values for LCT-API_AtndGroupAccessCodeType

* these values were never used!!

* 600	ATND	ATND	(duplicate of i80)
* t96	ANS	ANS	
*	GENANS	GENANS	(duplictate of n15)
* t98	REMOTE	REMOTE	

Values for LCT-API_AutocallOverride

u00	No Auto Call	Kein Direktruf
u01	No Override	Kein Aufschalten
u02	Override Only	Nur Aufschalten

u03	Override and Modifiable	Aufschalt- und Ã„nderbar

Values for LCT-API_BaudRate

*	AUTO	AUTO	(duplicate of t90)
u05	BAUD_RATE_300	BAUD_RATE_300	
u06	BAUD_RATE_1200	BAUD_RATE_1200	
u07	BAUD_RATE_2400	BAUD_RATE_2400	
u08	BAUD_RATE_4800	BAUD_RATE_4800	
u09	BAUD_RATE_9600	BAUD_RATE_9600	
u10	BAUD_RATE_19200	BAUD_RATE_19200	

Values for LCT-API_BitPattern

u13	ASCII	ASCII
u14	EBC	EBC
u15	IA2	IA2

Values for LCT-API_Block_SubAction

u16	CHANGE	CHANGE
u17	DELETE	DELETE
u18	INSERT	INSERT

Values for LCT-API_BoardName

*	PSIO	PSIO	(duplicate of)
*	SLMA	SLMA	(duplicate of 300)
*	SLMA1	SLMA1	(duplicate of 320)
*	SLMA2	SLMA2	(duplicate of 358)
*	SLMB	SLMB	(duplicate of 301)
u22	SLMB4	SLMB4	
u23	SLMB16	SLMB16	
u24	SLMO24	SLMO24	
u25	SLMQ16	SLMQ16	

*	SLMQ3	SLMQ3	(duplicate of 357)
u27	SLMQA	SLMQA	
*	SLMR	SLMR	(duplicate of 311)
u29	SLMR2	SLMR2	
u30	SLMR3	SLMR3	
*	SLMS	SLMS	(duplicate of 305)
u32	TMC16	TMC16	
u33	TMDID	TMDID	
u34	TMCOT	TMCOT	
u35	TMEMW	TMEMW	
u36	TMEMUS	TMEMUS	
u37	TMGSR	TMGSR	
u38	TMLSF	TMLSF	
u39	WAML	WAML	
u40	SIUP2	SIUP2	
u41	SIUX	SIUX	
u42	HORG	HORG	
u43	RG	RG	
*	STMA	STMA	(duplicate of 342)
*	TMD24	TMD24	(duplicate of 316)
u46	TMDN	TMDN	
u47	TMDN64	TMDN64	
u48	SLMN	SLMN	
u49	DMP5L	DMP5L	
u50	LTUC	LTUC	
u51	LTUCE	LTUCE	
u52	SCC	SCC	
*	IP	IP	(duplicate of)

*	ATI	ATI	(duplicate of 312)
u53	CCG	CCG	
*	CONF	CONF	(duplicate of 864)
u54	COT	COT	
*	DC	DC	(duplicate of)
u55	DCL	DCL	
u56	DDCL	DDCL	
u57	DID8	DID8	
u58	DIUR1	DIUR1	
u59	DIUR2	DIUR2	
*	DM3L	DM3L	(duplicate of)
u60	DM80M16	DM80M16	
u61	DM80M24	DM80M24	
u62	DM80M32	DM80M32	
*	DMP	DMP	(duplicate of t72)
u63	DP3DM	DP3DM	
u64	DP386	DP386	
u65	DP4L	DP4L	
u66	DPM	DPM	
*	FCD	FCD	(duplicate of)
*	FD	FD	(duplicate of)
*	HD	HD	(duplicate of)
u67	HD200	HD200	
u68	HD300	HD300	
u69	HD700	HD700	
*	HUBC	HUBC	(duplicate of)
u70	IOCC	IOCC	
u71	IOCG	IOCG	
u72	IOCSN	IOCSN	
*	IOPA	IOPA	(duplicate of)

u73	IOPAX	IOPAX	
*	IOPS	IOPS	(duplicate of)
u74	IPDN	IPDN	
*	LBC	LBC	(duplicate of)
*	LBU	LBU	(duplicate of)
*	LCC	LCC	(duplicate of)
*	LCR	LCR	(duplicate of)
u75	LCS0-D	LCS0-D	
*	LCU	LCU	(duplicate of)
u76	LTUR1	LTUR1	
u77	LTUR2	LTUR2	
u78	M2M	M2M	
u79	M2ML	M2ML	
u80	M8M	M8M	
*	MAC	MAC	(duplicate of)
u81	MBU	MBU	
u82	MD	MD	
*	MIP	MIP	(duplicate of)
u83	MIP2	MIP2	
u84	MTS	MTS	
u85	MTS64	MTS64	
u86	MTS/CONF	MTS/CONF	
u87	MTSCG	MTSCG	
u88	NCCG	NCCG	
u89	NCCG-X1	NCCG-X1	
u90	NCG	NCG	
u91	NCG-X100	NCG-X100	
*	OPS	OPS	(duplicate of 313)
u92	PCG	PCG	
u93	PHU	PHU	
u94	PSUB	PSUB	

u95	PSUC	PSUC	
u96	PSUD	PSUD	
u97	PSUF	PSUF	
u98	QDCL	QDCL	
*	RAX	RAX	(duplicate of)
*	RLI	RLI	(duplicate of 315)
u99	SICOE	SICOE	
v00	SIU	SIU	
v01	SIUCO	SIUCO	
v02	SIUCX	SIUCX	
v03	T1DN	T1DN	
v04	TTI	TTI	
*	VCD	VCD	(duplicate of)
*	DM4L	DM4L	(duplicate of)
*	DM80E	DM80E	(duplicate of)
*	DSC80	DSC80	(duplicate of)
*	SLMD	SLMD	(duplicate of 302)
v05	DP4PL	DP4PL	
v06	SLMA2P	SLMA2P	
v07	TMC16P	TMC16P	
v08	TMDN64P	TMDN64P	
*	SIUX2	SIUX2	(duplicate of 384)

Values for LCT-API_BoardType

v10	NOT_DEFINED	NOT_DEFINED	
*	PSIO	PSIO	(duplicate of)
*	SLMA	SLMA	(duplicate of)
v11	SLMA24	SLMA24	
*	SLMB	SLMB	(duplicate of)
*	SLMB16	SLMB16	(duplicate of)

*	SLMD	SLMD	(duplicate of)
v12	SLMPX	SLMPX	
*	SLMO24	SLMO24	(duplicate of)
*	SLMQ	SLMQ	(duplicate of)
*	SLMQ3	SLMQ3	(duplicate of)
*	SLMR	SLMR	(duplicate of)
v13	SLMR24	SLMR24	
*	SLMS	SLMS	(duplicate of)
*	TMC16	TMC16	(duplicate of)
*	TMCOT	TMCOT	(duplicate of)
*	TMDID	TMDID	(duplicate of)
*	TMEMW	TMEMW	(duplicate of)
*	TMGSR	TMGSR	(duplicate of)
*	TMLSF	TMLSF	(duplicate of)
*	SIUP2	SIUP2	(duplicate of)
*	SIUX	SIUX	(duplicate of)
*	RG	RG	(duplicate of)
*	STMA	STMA	(duplicate of)
v14	TMDBOS	TMDBOS	
v15	TMDMOS	TMDMOS	
*	TMDN	TMDN	(duplicate of)
*	SLMN	SLMN	(duplicate of)
*	DMP5L	DMP5L	(duplicate of)
*	LTUC	LTUC	(duplicate of)
*	LTUCE	LTUCE	(duplicate of)

Values for LCT-API_BusType

v18	FUNKT_CORNET	FUNKT_CORNET
v19	NUR_FUNKT	NUR_FUNKT
v20	NUR_CORNET_T	NUR_CORNET_T
v21	PUNKT_PUNKT_NW	PUNKT_PUNKT_NW
v22	PUNKT_PUNKT_COR	PUNKT_PUNKT_COR

v23	PUNKT_PUNKT_S0	PUNKT_PUNKT_S0	
v24	PUNKT_PUNKT_S2	PUNKT_PUNKT_S2	
v25	FUNKT_MIT_EAZ	FUNKT_MIT_EAZ	
v26	S0_NI	S0_NI	
*	NOT_DEFINED	NOT_DEFINED	(duplicate of)
v27	ONLY_SYMPHONY	ONLY_SYMPHONY	
v28	SYMPHONY_CORNET	SYMPHONY_CORNET	
v29	SYMPHONY_FUNCT	SYMPHONY_FUNCT	
v30	SYM_CORNET_FUNCT	SYM_CORNET_FUNCT	

Values for LCT-API_CP_DeviceType

v33	NULL_DEVICE	NULL_DEVICE	
v34	ANALOG	ANALOG	
v35	KEYSYSTEM	KEYSYSTEM	
*	FAX	FAX	(duplicate of)
v37	ATTENDANT	ATTENDANT	
v38	PHANTOM	PHANTOM	
v39	FICTICIOUS	FICTICIOUS	
v40	DCI	DCI	
v41	CTE	CTE	
v42	DATA_TERM	DATA_TERM	
v43	DIGITE_MAIN	DIGITE_MAIN	
v44	ATT_MAIN	ATT_MAIN	
v45	ATT_PERSONAL	ATT_PERSONAL	
v46	ATT_INTERNAL	ATT_INTERNAL	
v47	ATT_EXCHANGE	ATT_EXCHANGE	
v48	ATT_CALLPARK	ATT_CALLPARK	
v49	ATT_CHECK	ATT_CHECK	
v50	ATT_CHARGE	ATT_CHARGE	
v51	ATT_OUTGOING	ATT_OUTGOING	
v52	ATT_GENERAL	ATT_GENERAL	
v53	DCO_TERM	DCO_TERM	

v54	MAIN_SECR	MAIN_SECR
v55	PARC_LOCATION	PARC_LOCATION
v56	SB_FCT_TERM	SB_FCT_TERM

Values for LCT-API_CableType

v59	CABLE1	CABLE1
v60	CABLE2	CABLE2
xv61	CABLE3	CABLE3
v62	CABLE4	CABLE4
v63	CABLE5	CABLE5
v64	CABLE6	CABLE6
v65	CABLE7	CABLE7
v66	CABLE8	CABLE8
v67	CABLE9	CABLE9
v68	CABLE10	CABLE10
v69	CABLE11	CABLE11
v70	CABLE12	CABLE12
v71	CABLE13	CABLE13
v72	CABLE14	CABLE14
v73	CABLE15	CABLE15
v74	CABLE16	CABLE16
v75	CABLE17	CABLE17
v76	CABLE18	CABLE18
v77	CABLE19	CABLE19
v78	CABLE20	CABLE20
v79	CABLE21	CABLE21
v80	CABLE22	CABLE22
v81	CABLE23	CABLE23
v82	CABLE24	CABLE24
v83	CABLE25	CABLE25
v84	CABLE26	CABLE26

v85	CABLE27	CABLE27
v86	CABLE28	CABLE28
v87	CABLE29	CABLE29
v88	CABLE30	CABLE30
v89	CABLE31	CABLE31
v90	CABLE32	CABLE32

Values for LCT-API_CallByNameType

v93	NOT_USED	NOT_USED
v94	DATA_GRP_NAME	DATA_GRP_NAME
v95	NETWORK_ACCESS_NAME	NETWORK_ACCESS_NAME

Values for LCT-API_CallProgressState

v98	SUFFIX_DIALING	SUFFIX_DIALING
v99	IN_HOUSE_TO_ATND	IN_HOUSE_TO_ATND
w00	CO_TO_ATND	CO_TO_ATND
w01	PREFIX_DIALING_ATND	PREFIX_DIALING_ATND
w02	PREFIX_DIALING_STN	PREFIX_DIALING_STN

Values for LCT-API_CfwdDestType

*	UNKNOWN	UNKNOWN	(duplicate of p59)
w06	STNO	STNO	
w07	HUNT_GROUP	HUNT_GROUP	
w08	NETWORKING	NETWORKING	
w09	ACD	ACD	

Values for LCT-API_CircuitStatus

w12	Not in Service	Nicht in Betrieb
w13	Blocked	Blockiert
w14	Locked	Gesperrt
w15	Ready	Bereit
w16	Defect	Defekt
w17	Not Present	Nicht vorhanden

w18	Wrong Board	BG falsch
w19	Not Loaded	Nicht geladen
w20	In Use	Benutzt
w21	Seized	Belegt
w22	Reserved	reserviert

Values for LCT-API_CircuitUsage

w25	NO_CONNECTION	NO_CONNECTION
w26	ELEMENTARY_DEV	ELEMENTARY_DEV
w27	DEV_WITH_SU	DEV_WITH_SU
w28	NW_SU	NW_SU
w29	MULTIPLE_LINE_3	MULTIPLE_LINE_3
w30	BUS_WITH_CRI_DI_8	BUS_WITH_CRI_DI_8
w31	MULTIPLE_LINE_16	MULTIPLE_LINE_16
w32	MULTIPLE_LINE_30	MULTIPLE_LINE_30
w33	MULTIPLE_LINE_256	MULTIPLE_LINE_256
w34	AUN_PARKPLATZ_LINE	AUN_PARKPLATZ_LINE
w35	SIGNED_OFF	SIGNED_OFF

Values for LCT-API_ClockMode

w38	INTERNAL	INTERNAL	
*	EXTERNAL	EXTERNAL	(duplicate of)
w39	SLAVE	SLAVE	

Values for LCT-API_ConnectionType

w42	MULTI_LINE_8	MULTI_LINE_8	
w43	ELEMENTARY_DEVICE	ELEMENTARY_DEVICE	
w44	DEVICE_WITH_SUBUNITS	DEVICE_WITH_SUBUNITS	
*	NO_CONNECTION	NO_CONNECTION	(duplicate of)
*	SIGNED_OFF	SIGNED_OFF	(duplicate of)
w45	MULTI_LINE_3	MULTI_LINE_3	
w46	MULTI_LINE_16	MULTI_LINE_16	
w47	MULTI_LINE_30	MULTI_LINE_30	

w48	MULTI_LINE_256	MULTI_LINE_256
w49	NETWORK_SUBUNIT	NETWORK_SUBUNIT

Values for LCT-API_DCM_Type

w52	RMDCM	RMDCM
w53	ARMDCM	ARMDCM
w54	ADCM	ADCM
w55	DCMI	DCMI
w56	DCMII	DCMII
w57	DSDCO	DSDCO
w58	RPDCO	RPDCO
w59	WSDCO	WSDCO

Values for LCT-API_DMS_DeviceType

*	NONE	NONE	(duplicate of j48)
w63	HARD_DISK	HARD_DISK	
w64	FLPY_DRIV	FLPY_DRIV	
w65	CART_DRIV	CART_DRIV	
w66	MAGN_TAPE	MAGN_TAPE	
w67	DATREC	DATREC	

Values for LCT-API_DataBits

w70	DATA_BITS_5	DATA_BITS_5
w71	DATA_BITS_6	DATA_BITS_6
w72	DATA_BITS_7	DATA_BITS_7
w73	DATA_BITS_8	DATA_BITS_8

Values for LCT-API_DataInterface

w76	NULL_DATA	NULL_DATA
w77	AVD	AVD
w78	DCM	DCM
w79	TA_API	TA_API
w80	TA_232	TA_232
w81	TA_S0	TA_S0

*	DCI	DCI	(duplicate of)

Values for LCT-API_DataLineType

w84	TERMINAL	TERMINAL	
w85	COMPUTER	COMPUTER	
w86	DATAPOOL	DATAPOOL	

Values for LCT-API_DeviceAttached

w89	Optiset E Master Telephone	Optiset E Master Telefon	
w90	Optiset E Slave Telephone	Optiset E Slave Telefon	
w91	Optiset E Adapter (ISDN or Data)	Optiset E Adapter (ISDN oder Daten)	
w92	Optiset E Analog Adapter	Optiset E Analog Adapter	
w93	Analog Device on Optiset E Adapter (ISDN or Data)	Analog Device on Optiset E Adapter (ISDN oder Daten)	
w94	Optiset E NI-1200	Optiset E NI-1200	
*	Optiset E NI-1200	Optiset E NI-1200	(duplicate of)
w95	Optiset E NI-1200 TA AI	Optiset E NI-1200 TA AI	
*	Optiset E NI-1200 TA AI	Optiset E NI-1200 TA AI	(duplicate of)

Values for LCT-API_DeviceType

*	NULL_DEVICE	NULL_DEVICE	(duplicate of)
*	ANATE	ANATE	(duplicate of)
*	ATTENDANT	ATTENDANT	(duplicate of)
*	AVD	AVD	(duplicate of)
*	DYAD	DYAD	(duplicate of)
*	DYADJr	DYADJr	(duplicate of)
w98	EXT_VOICEMAIL	EXT_VOICEMAIL	
w99	EXEC_ICOM	EXEC_ICOM	
*	FAX	FAX	(duplicate of)
x00	IS_INTERFACE	IS_INTERFACE	

x01	MTA	MTA	
x02	MTA_EXP	MTA_EXP	
x03	NIGHT_CHIME	NIGHT_CHIME	
*	Off-Premise T1	Off-Premise T1	(duplicate of s15)
*	Off-Premise	Off-Premise	(duplicate of s14)
*	Analog	Analog	(duplicate of s10)
x04	PHONEMAIL	PHONEMAIL	
x05	PHONEMAIL_EXP	PHONEMAIL_EXP	
*	PHANTOM	PHANTOM	(duplicate of)
x06	PM_ACCESS	PM_ACCESS	
*	RP120	RP120	(duplicate of)
*	RP240	RP240	(duplicate of)
*	RP312	RP312	(duplicate of)
*	RP400	RP400	(duplicate of)
*	RP612	RP612	(duplicate of)
*	RP624	RP624	(duplicate of)
x07	RP4327	RP4327	
x08	ASSOC_DCI	ASSOC_DCI	
x09	SA_DCI	SA_DCI	
x10	ASSOC_DCM	ASSOC_DCM	
x11	SA_DCM	SA_DCM	

x13	NI1200	NI1200	
x14	NI_DATA	NI_DATA	

Values for LCT-API_DeviceUsage

x17	DCE	DCE	
x18	DTE_ModemPool	DTE_ModemPool	
x19	DTE_CodeConversionPool	DTE_CodeConversionPool	
*	MTA	MTA	(duplicate of)

x20	Cornet_T	Cornet_T	
x21	DCO	DCO	
*	DCM	DCM	(duplicate of)
x22	DCM_CodeConversionPool	DCM_CodeConversionPool	
x23	DCM_ModemPool	DCM_ModemPool	
x24	NationalISDN	NationalISDN	

Values for LCT-API_DialType

*	DTMF	DTMF	(duplicate of)
x26	Dial Pulsing	Impuls-Wahlverfahren	

Values for LCT-API_DigiteType

x29	DB_AM_DIGITE_211	DB_AM_DIGITE_211
x30	DB_AM_DIGITE_260	DB_AM_DIGITE_260
x31	DB_AM_DIGITE_SET400	DB_AM_DIGITE_SET400
x32	DB_AM_DIGITE_SET400_ST	DB_AM_DIGITE_SET400_ST
x33	DB_AM_DIGITE_SET400_C1	DB_AM_DIGITE_SET400_C1
x34	DB_AM_DIGITE_SET400_C2	DB_AM_DIGITE_SET400_C2
x35	DB_AM_DIGITE_SET500	DB_AM_DIGITE_SET500
x36	DB_AM_DIGITE_DYAD_JR	DB_AM_DIGITE_DYAD_JR
x37	DB_AM_DIGITE_DYAD	DB_AM_DIGITE_DYAD
x38	DB_AM_DIGITE_RP120	DB_AM_DIGITE_RP120
x39	DB_AM_DIGITE_RP240	DB_AM_DIGITE_RP240
x40	DB_AM_DIGITE_RP400PLUS	DB_AM_DIGITE_RP400PLUS
x41	DB_AM_DIGITE_NO_DIG	DB_AM_DIGITE_NO_DIG
x42	DB_AM_DIGITE_SYMPHONY	DB_AM_DIGITE_SYMPHONY
x43	DB_AM_DIGITE_NI_VOICE	DB_AM_DIGITE_NI_VOICE

Values for LCT-API_DyadType

*	DYAD	DYAD	(duplicate of)
*	DYADJr	DYADJr	(duplicate of)

Values for LCT-API_FramingMode

x46	STD	STD
x47	ESF	ESF

Values for LCT-API_HW_Assembly

x50	HW_NULL	HW_NULL
x51	NON_COMPACT	NON_COMPACT
x52	COMPACT	COMPACT
x53	EXT_COMPACT	EXT_COMPACT
x54	EXT_COMPACT_X	EXT_COMPACT_X
x55	EXT_COMPACT_DSC	EXT_COMPACT_DSC

Values for LCT-API_HeadsetType

x57	Modular	Modular
x58	Jack	Stecker

Values for LCT-API_Hicom_Type

x60	HICOM_NULL	HICOM_NULL
x61	HICOM_80	HICOM_80
x62	HICOM_180	HICOM_180
x63	HICOM_300	HICOM_300
x64	HICOM_3000	HICOM_3000
x65	HICOM_3000E	HICOM_3000E

Values for LCT-API_HuntSearchType

x68	Linear	Linear
x69	Circular	Zyklisch
x70	UCD	UCD

Values for LCT-API_HwyAccess

x73	HWY_ACC_NONE	HWY_ACC_NONE
x74	HWY_ACC_2	HWY_ACC_2
x75	HWY_ACC_4	HWY_ACC_4

Values for LCT-API_IncomingLinePreference

x77	Ring	Ruf
x78	Alert	Aufmerksamkeitsruf

Values for LCT-API_InvalidStnHuntCond

x80	OK_TO_STN_HUNT	OK_TO_STN_HUNT
x81	ITR_BLOCK	ITR_BLOCK
x82	ORIGINATE_ONLY	ORIGINATE_ONLY
x83	DICTATION_DEVICE	DICTATION_DEVICE
x84	EXECUTIVE_ICOM	EXECUTIVE_ICOM
x85	PAGING_DEVICE	PAGING_DEVICE
x86	ATTENDANT_CONSOLE	ATTENDANT_CONSOLE
x87	DEV_NOT_EXIST	DEV_NOT_EXIST
x88	DEV_CANNOT_STN_HUNT	DEV_CANNOT_STN_HUNT

Values for LCT-API_KLT_Type

x91	DB_AM_KLT_DIG211	DB_AM_KLT_DIG211
x92	DB_AM_KLT_DIG260	DB_AM_KLT_DIG260
x93	DB_AM_KLT_SET4	DB_AM_KLT_SET4
x94	DB_AM_KLT_SET4_STD	DB_AM_KLT_SET4_STD
x95	DB_AM_KLT_SET4_C1	DB_AM_KLT_SET4_C1
x96	DB_AM_KLT_SET4_C2	DB_AM_KLT_SET4_C2
x97	DB_AM_KLT_SET500	DB_AM_KLT_SET500
x98	DB_AM_KLT_SET500_F_PHONE	DB_AM_KLT_SET500_F_PHONE
x99	DB_AM_KLT_SET500_D_F_PHONE	DB_AM_KLT_SET500_D_F_PHONE
y00	DB_AM_KLT_SET500_KEYSET	DB_AM_KLT_SET500_KEYSET
y01	DB_AM_KLT_RP120	DB_AM_KLT_RP120
y02	DB_AM_KLT_RP312	DB_AM_KLT_RP312
y03	DB_AM_KLT_RP612	DB_AM_KLT_RP612
y04	DB_AM_KLT_RP240	DB_AM_KLT_RP240
y05	DB_AM_KLT_RP624	DB_AM_KLT_RP624
y06	DB_AM_KLT_RP400_PLUS	DB_AM_KLT_RP400_PLUS
y07	DB_AM_KLT_RP612K	DB_AM_KLT_RP612K

y08	DB_AM_KLT_RP624K	DB_AM_KLT_RP624K
y09	DB_AM_KLT_DYAD	DB_AM_KLT_DYAD
y10	DB_AM_KLT_SYMPHONY_T3	DB_AM_KLT_SYMPHONY_T3
y11	DB_AM_KLT_SYMPHONY_T8	DB_AM_KLT_SYMPHONY_T8
y12	DB_AM_KLT_SYMPHONY_T12	DB_AM_KLT_SYMPHONY_T12
y13	DB_AM_KLT_SYMPHONY_ADD_ON	DB_AM_KLT_SYMPHONY_ADD_ON
y14	DB_AM_KLT_NI1200	DB_AM_KLT_NI1200
y15	DB_AM_KLT_LODESTAR	DB_AM_KLT_LODESTAR
y16	DB_AM_KLT_LSTAR_ADDON	DB_AM_KLT_LSTAR_ADDON
y17	DB_AM_KLT_NI_VOICE	DB_AM_KLT_NI_VOICE

Values for LCT-API_KeyFnctType

y20	ACCT	ACCT	
y21	ACDPQ	ACDPQ	
y22	ACDPS	ACDPS	
y23	ACDSQ	ACDSQ	
y24	ACDSS	ACDSS	
*	ACK	ACK	(duplicate of)
y25	AGMSG	AGMSG	
*	AUTOM	AUTOM	(duplicate of)
y26	AUTOMLOC	AUTOMLOC	
y27	AVAIL	AVAIL	
y28	BADLN	BADLN	
*	BROADCST	BROADCST	(duplicate of)
*	BUZZ	BUZZ	(duplicate of)
y29	CALLWT	CALLWT	
*	CBK	CBK	(duplicate of)
y30	CLEAR	CLEAR	
*	CNCT	CNCT	(duplicate of)
*	CONF	CONF	(duplicate of)
*	DATA	DATA	(duplicate of D13)
y32	DATALOC	DATALOC	

y33	DDS	DDS	
y34	DDSLOC	DDSLOC	
*	DSS	DSS	(duplicate of)
y35	DTS	DTS	
y36	EMGCY	EMGCY	
y37	ENTID	ENTID	
y38	EXHLD	EXHLD	
*	FWD	FWD	(duplicate of)
*	HEADSET	HEADSET	(duplicate of)
y39	HOLDM	HOLDM	
y40	HOLDMLOC	HOLDMLOC	
y41	ICOM	ICOM	
y42	INTCM	INTCM	
*	IUSE	IUSE	(duplicate of)
*	LINE	LINE	(duplicate of)
y43	LINETAA	LINETAA	
*	LNR	LNR	(duplicate of)
*	MB	MB	(duplicate of)
*	MENU	MENU	(duplicate of)
*	MUTE	MUTE	(duplicate of)
y44	MUTELOC	MUTELOC	
y45	OVR	AUFSCHAL	
y46	PHML.	PHML.	
y47	PREVIEW.	PREVIEW.	
y48	PRIV	PRIV	
y49	PROG	PROG	
*	PU	PU	(duplicate of)
y50	RFLASH	RFLASH	
y51	RCOFF	RCOFF	
y52	RPTID.	RPTID.	
*	SCROL	SCROL	(duplicate of)
y53	SILMT	SILMT	

*	SNR	SNR	(duplicate of)
*	SPARK	SPARK	(duplicate of)
*	SPKR	SPKR	(duplicate of)
*	SPKRCALL	SPKRCALL	(duplicate of)
y54	SPKRLOC	SPKRLOC	
y55	SPMSG	SPMSG	
*	ST	ST	(duplicate of)
* y56 SPDI	(duplicate of k05)	(duplicate of k05)	
y57	STORE	STORE	
y58	STPHNT	STPHNT	
*	SYSSP1	SYSSP1	(duplicate of)
*	SYSSP2	SYSSP2	(duplicate of)
*	TIME	TIME	(duplicate of)
y59	TOGGL	TOGGL	
y60	TONMT	TONMT	
y61	UNAVA	UNAVA	
y62	VACANT	VACANT	
*	VC	VC	(duplicate of)
y63	VCALL	VCALL	
*	VCR	VCR	(duplicate of)
y64	VOLUP	VOLUP	
y65	VOLDN	VOLDN	
y66	WORK	WORK	
y67	XFER	XFER	

Values for LCT-API_KeyLayoutType

*	RP120	RP120	(duplicate of)
*	RP312	RP312	(duplicate of)
*	RP612	RP612	(duplicate of)
*	RP240	RP240	(duplicate of)
*	RP624	RP624	(duplicate of)
*	RP400	RP400	(duplicate of)

*	RP612K	RP612K	(duplicate of)
*	RP624K	RP624K	(duplicate of)
*	DYAD	DYAD	(duplicate of)
*	OPTIE3	OPTIE3	(duplicate of)
*	OPTIE8	OPTIE8	(duplicate of)
*	OPTIE12	OPTIE12	(duplicate of)
*	OPTIKEYM	OPTIKEYM	(duplicate of)
*	NI1200	NI1200	(duplicate of)

Values for LCT-API_KeyModuleType

y70	BASE	BASE
y71	KEYMODULE1	KEYMODULE1
y72	KEYMODULE2	KEYMODULE2
y73	KEYMODULE3	KEYMODULE3
y74	KEYMODULE4	KEYMODULE4

Values for LCT-API_LCRCOS_AlternateAdvanceTimer

y76	0 seconds	0 Sekunde
y77	2 seconds	2 Sekunden
y78	5 seconds	5 Sekunden
y79	10 seconds	10 Sekunden
y80	15 seconds	15 Sekunden

Values for LCT-API_LCRCOS_Type

y82	LCRCOS_TYPE_VOICE	LCRCOS_TYPE_VOICE
y83	LCRCOS_TYPE_DATA	LCRCOS_TYPE_DATA

Values for LCT-API_LanguageType

*	German	German	(duplicate of)
*	English	English	(duplicate of)
*	French	French	(duplicate of)
*	Spanish	Spanish	(duplicate of)
*	Portugues	Portugues	(duplicate of)
*	Dutch	Dutch	(duplicate of)

*	Danish	Danish	(duplicate of)
*	Finnish	Finnish	(duplicate of)
*	Greek	Greek	(duplicate of)
*	Italian	Italian	(duplicate of)

Values for LCT-API_LineDistance

*	Less than 100	Less than 100	(duplicate of)
*	Between 100 and 1000	Between 100 and 1000	(duplicate of)
*	Greater than 1000	Greater than 1000	(duplicate of)

Values for LCT-API_ListType

y85	BOARD	BOARD	
y86	SMALL_DEVICE	SMALL_DEVICE	
y87	DEVICE	DEVICE	
y88	EXTEND_DEVICE	EXTEND_DEVICE	
y89	XREF	XREF	
y90	ATND_GROUP	ATND_GROUP	
y91	COM_GROUP	COM_GROUP	
y92	CONNECT_GROUP	CONNECT_GROUP	
*	HUNT_GROUP	HUNT_GROUP	(duplicate of)
y93	PICKUP_GROUP	PICKUP_GROUP	
y94	DATA_POOL	DATA_POOL	
y95	IS_POOL	IS_POOL	
*	ICOM	ICOM	(duplicate of)
*	COS	COS	(duplicate of)
y96	COSX	COSX	
y97	LCRCOS	LCRCOS	
y98	STD_KEY_LAYOUT	STD_KEY_LAYOUT	
y99	EPNP_DPLN	EPNP_DPLN	
z00	LOCAL_DPLN	LOCAL_DPLN	
z01	DOWNLOAD_IDX	DOWNLOAD_IDX	
z02	HOTLINE_IDX	HOTLINE_IDX	
z03	NIGHT_OPTION	NIGHT_OPTION	

z04	SPEEDDIAL	SPEEDDIAL

Values for LCT-API_NI_DataAPI_Link

z06	API_LINK_OFF	API_LINK_OFF
z07	API_LINK_ON	API_LINK_ON

Values for LCT-API_NI_DataMultlink

z09	MULTLINK_OFF	MULTLINK_OFF
z10	MULTLINK_ON	MULTLINK_ON

Values for LCT-API_NW_BalanceNum

z12	Test 600 Ohm	Test 600 Ohm
z13	Compnet	Compnet
z14	Long loop A	Long loop A
z15	Long loop B	Long loop B

Values for LCT-API_NetUser

z17	USER	USER
z18	NETWK	NETWK

Values for LCT-API_NightOptionType

*	STN	STN	(duplicate of)
*	NONE	NONE	(duplicate of)
*	GENANS	GENANS	(duplicate of)
*	REMOTE	REMOTE	(duplicate of)

Values for LCT-API_NonCfwdDest

z20	DEST_OK_TO_CFWD	DEST_OK_TO_CFWD
z21	DEST_ITR	DEST_ITR
z22	DEST_CANNOT_RECEIVE_CALLS	DEST_CANNOT_RECEIVE_CALLS
z23	DEST_DICTATION_DEVICE	DEST_DICTATION_DEVICE
z24	DEST_EXECUTIVE_ICOM	DEST_EXECUTIVE_ICOM
z25	DEST_PAGING_DEVICE	DEST_PAGING_DEVICE
z26	DEST_CANNOT_CFWD	DEST_CANNOT_CFWD

Values for LCT-API_OffType

z28	IMMEDIATE	IMMEDIATE	
z29	CONDITIONAL	CONDITIONAL	
z30	LONG_TERM	LONG_TERM	

Values for LCT-API_OnOff

*	OFF	OFF	(duplicate of)
z31	ON	ON	

Values for LCT-API_OnType

z33	UNCONDITIONAL	UNCONDITIONAL	
*	CONDITIONAL	CONDITIONAL	(duplicate of)

Values for LCT-API_OperationMode

z34	Demand	Anforderung
z35	Continuous	Lückenlos

Values for LCT-API_OperationType

z36	Fixed	Fest
z37	Random	Stichprobenweise

Values for LCT-API_OptisetE_Type

z39	Entry	Eintrag
z40	Basic	Basis
z41	Standard	Standard
z42	Advance	Advance
z43	Advance Plus	Advance Plus
z44	Speakerphone	Speakerphone
z45	Full Duplex Phone Std	Full Duplex Phone Std
z46	Full Duplex Phone Adv Plus	Full Duplex Phone Adv Plus
z47	UP0/E PC Phone	UP0/E PC Phone
z48	Dig. Wireless Phone Adapt.	Dig. Wireless Phone Adapt.
z49	Reserved for new phone (1)	Reserviert für neues Telefon (1)
z50	Reserved for new phone (2)	Reserviert for neues Telefon (2)
z51	Digital Cordless Phone	Digital Cordless Phone

Values for LCT-API_OutgoingLinePreference

* z52	Last	Last	(duplicate of)
* z53	Primary	Primary	(duplicate of H30)
z54	Idle	Ruhe	

Values for LCT-API_Parity

z56	ODD	ODD	
z57	EVEN	EVEN	
*	NONE	NONE	(duplicate of)
z58	MARK	MARK	
z59	SPACE	SPACE	

Values for LCT-API_PartNumber (continuation!)

z62	Q2161-X100	Q2161-X100
z63	Q2271-X100	Q2271-X100
z64	Q2187-X	Q2187-X
z65	Q2168-X	Q2168-X

10.3.116 Codes for port, reg_type

D48	Dynamic	Dynamisch
D49	Static	Statis

10.3.117 Opera Addon type

544	1 OPERA KEY MOD	1 OPERA KEY MOD
545	2 OPERA KEY MOD	2 OPERA KEY MOD
549	OPERA UP0 EXTENDER	OPERA UP0 EXTENDER

544	1 OPENSTAGE KEY MOD	1 OPENSTAGE KEY MOD
545	2 OPENSTAGE KEY MOD	2 OPENSTAGE KEY MOD
549	OPENSTAGE UP0 EXTENDER	OPENSTAGE UP0 EXTENDER

981	1 OPERA LOW COST KEY MOD	1 OPERA LOW COST KEY MOD

982	1 OPERA BLF KEY MOD	1 OPERA BLF KEY MOD
983	1 OPENSCAPE KEY MOD 55	1 OPENSCAPE KEY MOD 55
984	2 OPENSCAPE KEY MOD 55	2 OPENSCAPE KEY MOD 55
985	1 OPENSCAPE KEY MOD CP400	1 OPENSCAPE KEY MOD CP400
986	2 OPENSCAPE KEY MOD CP400	2 OPENSCAPE KEY MOD CP400
987	1 OPENSCAPE KEY MOD CP600	1 OPENSCAPE KEY MOD CP600
988	2 OPENSCAPE KEY MOD CP600	2 OPENSCAPE KEY MOD CP600
989	3 OPENSCAPE KEY MOD CP600	3 OPENSCAPE KEY MOD CP600
990	4 OPENSCAPE KEY MOD CP600	4 OPENSCAPE KEY MOD CP600
991	1 OPENSCAPE KEY MOD CP700	1 OPENSCAPE KEY MOD CP700
992	2 OPENSCAPE KEY MOD CP700	2 OPENSCAPE KEY MOD CP700
993	3 OPENSCAPE KEY MOD CP700	3 OPENSCAPE KEY MOD CP700
994	4 OPENSCAPE KEY MOD CP700	4 OPENSCAPE KEY MOD CP700

10.4 Error Texts for ODFs with Import Function

10.4.1 Error Texts (English)

Table 235: Errors of ODF BUILDING

7551	Name of building has to be entered.
7552	Name of location has to be entered.
7553	Name of location cannot be modified as it is already defined for subscribers.
7554	Location <location> does not exist.
7555	Building <building> in location <location> is already present.
7556	Building <building> in location <location> does not exist.

7557	Stations are still existing for building <building>. It cannot be deleted.

Table 236: Errors of ODF BUTTON_INDIV

7941	The individual key layout will be supported from HICOM V3.4 onwards
7942	The station <station> has either not been created, or cannot be specified as a direct station destination.
7943	The individual key layout can"t be modified, since neither DSS nor NAME key exists.
7944	The individual key layout cannot be changed for a dummy Cordless E station.
7945	In system <system> no more direct station selection keys (DSS) or name keys (NAME) can be added.
7946	DIGI211 Digital Device Type does not allow individual modifications of the key functions to be performed.
7947	A maximum of 28 <key> keys can be allocated.
7948	The following key functions for key system are only authorized for HICOM300E V3.0 and later: IUSE, LINE, EXHOLD, HOLD, PRIVACY, RCUTOFF, PRIM
7949	Station <station> is not defined as key system. Please use the field "Key System" in window "Station" to define it.
7950	Station <station> is defined as key system. Please use the field "Key System" in window "Station" to delete it.
7951	The secondary line call number <line> must differ from the personal call number.
7952	For the secondary line <line>, station<station> must be defined as Key System. Please use the field "Key System" in window "Station" to define it.
7953	The secondary line <line> is not existing or not defined as Key system.
7954	The secondary line <line> is already allocated to 40 Key System stations.
7955	Interface Error: Station <station> is not a Key System.
7956	Interface Error: Station <station> is a Key System.
7957	Allocation of line keys to secondary lines may only be done once.
7958	The ring option of the primary line must be set since the incoming line preference has the value <value>.
7959	Station <station> is or still a member of a networkwide team. Ring transfer and deputy keys cannot be administered.
7960	Station <station> is or still a member of a networkwide team. DSS destinations, primary and executive line %s cannot be modified.
7961	Station <station> is or still a member of a networkwide team. DSS and line keys cannot be administered.
7962	Station<station> is or still a member of a networkwide team. DSS destinations, primary and executive line %s cannot be administered.

7963	<line> secondary lines are or still are executive lines of a networkwide teams.

Table 237: Errors of COMPANY

7581	Country and company have to be entered.
7582	Name of company cannot be modified since subscribers are already registered for it.
7583	Company <company> in country <country> is already registered.
7584	Company <company> in country <country> is not present.

Table 238: Errors of COUNTRY

7611	Destination and originating country have to be entered.
7612	Area code has to be entered.
7613	An area code is already registered for country of origin <orig country> and destination country <dest country>.
7614	Destination country has to differ from country of origin.
7615	An area code is not defined for country of origin <orig country> and destination country <dest country>.

Table 239: Errors of DEVCONST

6651	Auxiliary devices 3 and 4 cannot be added or modified.
6652	Name of device combination has to be entered.
6653	Device class and type have to be entered.
6654	For card reader, hands-free talking and display, a value has to be entered.
6655	Since the device combination is still allocated to a station, only fields which contain no system data can be changed.
6656	<device> device combination already exists.
6657	Combination of device class <device class>, type <type>, add-on device <device>, adapter 1 <adapter> and adapter 2 <adapter> is not possible.
6658	Device type <device type> does not correspond to device class <device class>.
6659	Add-on device <device> is not possible with device class <device class>.
6660	Add-on device <device> is not possible with device type <device type>.
6661	<device combination> device combination not found.
6662	<device combination> device combination cannot be deleted since it is still allocated to one or more stations.
6663	not supported
6664	Device class<device class> with card reader allows a maximum of three add-on devices.

6665	Adapter arrangement is not allowed.
6666	Adapter <adapter> is not allowed with device class <device class>.
6667	Adapter <adapter> is not allowed with device type <device type>.
6668	The parameter "Hands-free" is not administered for device type OPTISET.
6669	For card reader and display, a value must be entered.
6670	Device type <device type> allows only one adapter.
6671	The parameters "ComManager", "Headset" and "Recorder" can only be entered for Optisets.
6672	An Optiset terminal device cannot be configured simultaneously with card reader and ComManager.
6673	An Optiset with ComManager requires 4 add-on devices.
6674	No other auxiliary devices can be entered or modified.
6675	Device class <device class> with card reader allows a maximum of one add-on device.
6676	An Optipoint 500 with ComManager requires 2 add-on devices.

Table 240: Errors of DISPLAYNAME

8181	The station <station> does not exist on the system <system>.
8182	Display name must not be specified for the <device combination> device combination. Action is not executable.
8183	The display name must not be longer than 30 characters.
8184	In system <system> no more display names can be added.
8185	The station <station> does not have a display name at position %s.
8186	The station <station> already has a display name at position %s.
8187	The maximum number of display names has been exceeded.
8188	In system <system> no more organizational units can be added.
8189	The <field> field will be supported from OpenScape 4000 onwards.
8190	The station <station> already has a display name.
8191	The station <station> is prepared for a move. Action is not executable.

Table 241: Errors of LOCATION

7521	Name of location has to be entered.
7522	Name of location cannot be modified since it is already defined for subscribers.
7523	Location <location> is already added.
7524	Location <location> is not present.

7525	Stations are still existing for location <location>. It cannot be deleted.

Table 242: Errors of ORG

7641	At least one of the organisation levels has to be entered.
7642	Company and country must be known.
7644	Country and company can not be modified.
7645	Company <company> in country <country> does not exist.
7649	Organization <organization> does not exist.
7650	Organization <organization> already exists.

Table 243: Errors of PERSDAT

7461	Name of subscriber has to be entered.
7462	Country of subscriber has to be entered.
7463	Location of building <building> has to be entered.
7464	No domain with tie line <line> exists.
7465	Subscriber <subscriber> is not present.
7466	System not found.
7467	For subscriber <subscriber> no more position can be found in system <system>.
7468	If an organization unit is allocated to the subscriber, the company has to be entered.
7469	The fields Pers-APD-Typ-TBkz and AKZ1 must be completed.
7470	The fields Country, Company and Location must be completed.
7471	The fields Department and Kst-Abtl. must be completed.
7472	The fields Service Type Prefix and Service Type (the first field at least) must be completed.
7473	The service type prefix must be exactly 7 characters long.
7474	The value <value> is invalid. Only 4-digit values are allowed.
7475	The server must be defined.
7476	The mailbox ID must be defined.
7477	The PIN must be defined.
7478	The server cannot be changed.
7479	The mailbox ID cannot be changed.
7480	The mailbox with ID <mailbox ID> is already existing.
7481	A mailbox with PIN <PIN> is already existing.
7482	You must first configure a mailbox for the station.

7483	The personal ID number must be defined.
7484	The personal ID number must be exactly 6-character long.
7485	The personal ID number is already assigned to another subscriber.
7486	Once a mailbox is configured, the system and extension must be defined.

Table 244: Errors of PIN

6540	Station with domain, system and extension has to be known.
6541	PIN type has to be entered.
6542	PIN has to be known.
6543	Station with domain, system and extension cannot be modified.
6544	PIN, PIN type, PIN class and individual PIN cannot be changed.
6545	PIN <PIN> is already used in system <system>.
6547	PIN <PIN> not found in system <system>.
6548	A PIN cannot be assigned for subscriber <subscriber> on system "<system> because it is not permitted by his terminal.
6549	No new PIN can be added because limit is reached in system <system>.
6550	No free PIN found.
6551	PIN <PIN> of PIN type <PIN type> has to be %s-characters long.
6552	PIN <PIN> cannot be deleted because it is used for mailbox of station <station>.
6553	PIN type <PIN type> is not allowed in system <system>.
6554	The position cannot be entered for the PIN type <PIN type>.
6555	The Cordless E PIN <PIN> is already assigned.
6556	The station <station> is not a mobile station.
6557	Subscriber <subscriber> already has the COS changeover code <COS changeover code> for PIN class <PIN class> on system <system>.
6558	The Cordless E PIN <PIN> is not permitted.
6559	Domain and system must be known.
6560	The PIN type <PIN type> must not be assigned to a subscriber.
6561	The PIN type or the PIN class must be entered.
6562	PIN type <PIN type> for PIN class <PIN class> is not permitted on system <system>.
6563	The PIN type for PIN class <PIN class> on system <system> cannot be found.
6564	The PIN class for PIN type <PIN type> on system <system> cannot be found or a unique instance of this PIN class cannot be found.

6565	Length entry cannot be changed - all pins of pin type <PIN type> must first be deleted.
6566	Pin Class <PIN class> does not exist on switch <switch>.

Table 245: Errors of PORT

4350	The <device combination> device combination is not supported for HICOM300E V3.0 or V3.1 system type.
4351	No further call logs can be created in system <system> for <device combination> device combination.
4352	The field <field> must not be set.
4353	No further application with workstation protocol (API) can be installed on system <system> with the <device combination> device combination.
4354	Interface Error: The Key System"s primary line does not exist.
4355	Interface Error: The Key System"s primary line exists.
4356	The key system"s standard key layout cannot be changed.
4357	Key system move is not permitted.
4358	Defining or deleting a key system is only allowed with OPTISET terminal device.
4359	Station <station> is defined as "secretary".
4360	The networkwide configuration of the key system is not possible.
4361	The <device combination> device combination cannot be defined. The logged out key system allows no ComManager.
4362	The <device combination> device combination cannot be defined. The key system needs an OPTISET terminal device.
4363	The <device combination> device combination cannot be defined. The line keys on the first add-on device are still allocated to secondary or primary lines.
4364	The PEN and connection type of the key system cannot be changed.
4365	On system <system> no further key system can be defined.
4366	The value <value> in field <field> is not permitted since the ring option of the primary line is not set.
4367	A line key is needed for creating the key system.
4368	The <device combination> device combination cannot be defined. The line keys on the base device are still allocated to secondary or primary lines.
4369	The <feature>, <feature> and <feature> features are not possible on the device combination <device combination>.
4370	The station <station> is a <station type> on an Executive/Secretary system. It cannot be defined as a key system.
4371	The <device combination> device combination cannot be created on a bus. An OPTISET line with DCI700 requires an S0 adaptor.

4372	Station <station> is or still a member of a networkwide team. Only some of the fields can be modified.
4373	Station <station> is or still a member of a networkwide team. The action can"t be executed.
4374	The <feature> and <feature> features must not be set.
4375	The <feature> feature must not be modified.
4376	The default key layout can"t be modified since <feature> or <feature> feature is set.
4377	Station <station> is a card reader for DSS (direct station selection) in the networkwide reserved <pickup group> pickup group.
4378	Device combination, connection type and TSI parameter can"t be modified since OPTISET extensions are still existing on the OPTISET station.
4379	The< DTB access> DTB access is only possible if the DTB service code is known at system <system>.
4380	Mobile station with data terminal device of type <device type> is not supported.
4381	The <parameter>, <parameter> and <parameter> parameters are necessary from Cordless E version 2.3 and later.
4382	The fixed call forwarding can"t be activated.
4383	The combination of parameter <parameter> = <parameter> and <parameter> = <parameter> is not allowed.
4384	The <device combination> device combination is not supported for OpenScape 4000 V1.0 system type.

6051	Extension has to be entered.
6052	Name of system has to be entered.
6053	Name of device combination has to be entered.
6054	You must enter the first class of service (COS 1).
6055	You must enter the second class of service (COS 2).
6056	Port equipment number (PEN) "%s" does not fit to "%s" device combination or it is only allowed for a station connected to a bus.
6057	STA extension "%s" is not consistent.
6058	In domain "%s" extension "%s" is not consistent.
6059	In system "%s" subscriber extension "%s" is not consistent.
6060	Standard key layout for "%s" device combination has to be entered.
6061	Standard key layout for "%s" device combination is not allowed to be entered.
6062	STA extension "%s" is not free.
6063	In domain "%s" STA extension "%s" is not free.
6064	At system "%s" STA extension "%s" is not free.

6065	Device combination "%s" requires parameter record for automatic downloading.
6066	Station "%s" does not exist at system "%s".
6067	Codes for voice call warning tone and callback rejection are not allowed to be entered.
6068	The bus parameters "connection type", "protocol", "TSI" and "option" are only necessary for a station on a bus.
6069	Device combination "%s" does not require connection type.
6070	During "move" within a domain extension number has to be modified.
6071	Least Cost Routing COS is not allowed to be entered.
6072	In system "%s" key layout "%s" does not fit to "%s" device combination. Check add-on device.
6073	"%s" is not a STA extension.
6074	In domain "%s", "%s" is not a STA extension.
6075	In system "%s", "%s" is not a STA extension.
6076	Extension "%s" does not exist.
6077	In domain "%s" extension "%s" does not exist.
6078	In system "%s" extension "%s does not exist.
6079	"%s" device combination without FAX device doesn"t require the parameters "Protocol", "TSI" and "Option".
6080	Domain for system "%s" not found.
6081	No free consistent STA extension found in system "%s".
6082	No free consistent STA extension found in domain "%s".
6083	"%s" device combination does not exist.
6084	PEN or connection type for "%s" device combination has to be entered.
6085	Indices to option table for functional devices are not allowed to be entered.
6086	Interface error: System and/or extension missing !
6087	All PINs cannot be accepted in system "%s".
6088	"%s" device combination without data device doesn"t require the parameters "Protocol", "TSI" and "Option".
6089	Protocol type of functional DTE device of "%s" device combination has to be entered.
6090	Protocol type of functional FAX device of "%s" device combination has to be entered.
6091	System "%s" is not supported for the "move".
6092	On OPTISET station with main extension "%s", two OPTISET extensions already exist.
6093	"%s" device combination is not supported for system type HICOM300 V3.2.
6094	"%s" device combination is not allowed for card subscribers.

6095	"%s" device combination is not supported for system type HICOM300 V3.3.
6096	Device type for "%s" device combination not found.
6097	Interface error: Name of domain missing !
6098	Port equipment number (PEN) "%s" is not free.
6099	Port equipment number (PEN) "%s" is not free in domain "%s".
6100	Port equipment number (PEN) "%s" is not free in system "%s".
6101	There is no system with port equipment number (PEN) "%s".
6102	In domain "%s" no system with port equipment number (PEN) "%s" exists.
6103	In system "%s" port equipment number (PEN) does not exist.
6104	No appropriate free port equipment number (PEN) found in system "%s".
6105	Action is not executable due to unrecoverable inconsistency of technical extension "%s" in system "%s".
6106	In domain "%s" no system with class of service "%s" exist.
6107	TSI number "%s" of voice device cannot be modified.
6108	The first speed dial list "%s" does not exist in domain "%s".
6109	The second speed dial list "%s" does not exist in domain "%s".
6110	In system "%s" key layout "%s" does not fit to "%s" device combination.
6111	In system "%s" no more S0 bus can be added.
6112	During "move" to another domain, extension is not allowed to be modified.
6113	Station "%s" is not prepared for a move.
6114	"%s" device combination does not require parameter for downloading.
6115	No port equipment number (PEN) is assigned to a card station.
6116	Stations cannot be changed to card stations.
6117	During "move" within a system, port equipment number (PEN) has to be modified.
6118	In system "%s" no more display names can be added.
6119	In system "%s" no more stations can be added.
6120	In system "%s" no more stations with individual speed dialing facility can be added.
6121	In system "%s" no more stations with "%s" device combination can be added.
6122	Station "%s" is a technical station.
6123	Station "%s" is prepared for a move.
6124	Action is not executable due to unrecoverable inconsistencies of STA extensions "%s" in domain "%s".

6125	In system "%s" download parameter class "%s" does not exist.
6126	In system "%s" download parameter record "%s" does not exist.
6127	Move" is not possible for mobile stations or stations on a bus or OPTISET port. There are bus extensions existing still.
6128	"%s" device combination without functional DTE device doesn"t require the parameters "Protocol" and "Option".
6129	A bus or OPTISET port with the main extension "%s" is not existing.
6130	No system with class of service (COS) "%s" exist.
6131	MSN extensions are configured on a S0 bus or an OPTISET port. The action can"t be executed.
6132	In system "%s" class of service (COS) "%s" does not exist.
6133	The action cannot be performed for the unsupported "%s" device combination or a subscriber with ACD properties or an ATM subscriber (TFA).
6134	A limited number of parameters can be modified for the unsupported "%s" device combination or a subscriber with ACD properties or an ATM subscriber (TFA).
6135	Basic speed dial list and speed dial list "%s" does not exist in system "%s".
6136	The first speed dial list "%s" does not exist.
6137	The first speed dial list "%s" does not exist in system "%s".
6138	The second speed dial list "%s" does not exist in system "%s".
6139	The second speed dial list "%s" does not exist.
6140	"%s" device combination can"t be defined on a bus. An OPTISET line requires: a/b adaptor for analogue device w/o protocol; S0 adaptor for DTE or alternatively, data adaptor for functional devices with protocol.
6141	"%s" device combination does not fit to port equipment number (PEN) "%s" and connection type "%s".
6142	There are still extensions on a bus port with the main extension number "%s".
6143	The device combination, PEN and the connection type cannot be changed since there are still extensions on the bus or OPTISET port.
6144	PEN "%s" doesn"t correspond to the PEN on the bus or OPTISET port with the main extension "%s".
6146	Connection type "%s" doesn"t correspond to the connection type on the bus or OPTISET port with main extension "%s".
6147	The PEN of the bus port cannot be found in system "%s" with the main extension "%s".
6148	Bus type on a S0 bus or an OPTISET port with main extension "%s" not found.
6149	The system cannot locate the connection type at the bus, functional S0/S2 or OPTISET port with the main extension "%s".
6150	The connection type and PEN cannot be changed on a bus or OPTISET extension.

6151	S0 bus with main extension "%s" cannot be extended with "%s" device combination.
6152	No system with download parameter class "%s" exists.
6153	Three bus extensions are already existing on the bus port with the main extension "%s".
6154	On the S0 bus, not more than 8 devices can be configured.
6155	TSI number "%s" of data terminal device cannot be modified.
6156	TSI number "%s" of FAX terminal device cannot be modified.
6157	TSI number "%s" of voice terminal device is already used.
6158	TSI number "%s" of data terminal device is already used.
6159	TSI number "%s" of FAX terminal device is already used.
6160	No system with download parameter record "%s" exists.
6161	Card stations are not allowed on S0 bus.
6162	Move of card stations is not allowed.
6163	In system "%s" no more direct station selection keys (DSS) or name keys (NAME) can be added.
6164	For extension "%s", a VMS mailbox already exists.
6165	Station "%s" is member of a night service. An administration in system "%s" is required.
6166	Station "%s" is still associated. An administration in system "%s" is required.
6167	Station "%s" is still used as analogue part of a modem. An administration in system "%s" is required.
6168	Station "%s" is still CLC capable. An administration in system "%s" is required.
6169	Card station cannot be modified.
6170	The extension number and PEN of the station prepared for a move cannot be modified.
6171	Extension and port equipment number (PEN) of technical station cannot be modified.
6172	"%s" device combination is not supported for system type HICOM V3.4.
6173	Least Cost Routing COS for data is not allowed to be entered.
6174	Least Cost Routing COS have to be entered.
6175	The "%s" device combination is not or only allowed for a station on a bus.
6176	System and domain cannot be modified.
6177	Extension cannot be modified.
6178	The PEN of the OPTISET port cannot be found in system "%s" with the main extension "%s".
6179	OPTISET port with main extension "%s" cannot be extended with "%s" device combination.
6180	On an OPTISET port, no more than three terminal devices can be configured.
6181	Fields used for display cannot be modified.

6182	On OPTISET port with main extension "%s" no slave phone adapter can be configured.
6183	On OPTISET port with main extension "%s", no S0 adapter can be configured.
6184	Log off of station or log on of card station is not allowed.
6185	Log on of card station is not allowed.
6186	Station or card station cannot be modified.
6187	Move of signed off stations or card stations is not allowed.
6188	Logged off station or card station are not allowed on a bus or OPTISET port.
6189	log off and log on of stations on S0 bus or OPTISET portis currently not possible.
6190	Device combination "%s" is not allowed for logged off or logged in station (no data terminal devices).
6191	No port equipment number (PEN) is assigned to a signed off station.
6192	A technical station cannot be signed off.
6193	A station prepared for a move cannot be signed off.
6194	Port equipment number (PEN) "%s" is not signed off.
6195	Port equipment number (PEN) "%s" is not signed off in domain "%s".
6196	Port equipment number (PEN) "%s" is not signed off in system "%s".
6197	A suitable signed off port equipment number (PEN) in system "%s" cannot be found.
6198	"%s" device combination is not allowed for signed on card stations.
6199	Processing HICOM data is still running. Please try again later or select once again.
6200	Call forward - all call destination can"t be one own"s station.
6201	The main extension cannot be modified.
6202	No further "call forward - all calls" can be created in the system "%s".
6203	Call forwarding must not be specified.
6204	OPTISET station with main extension "%s" and an adaptor extension cannot be extended with "%s" device combination.
6205	On an unsupported "%s" device combination, signed off stations or card stations cannot be taken in service.
6206	Sign off" of station with unsupported "%s" device combination is not allowed.
6207	The station "%s" is a member of an executive/secretary group. The "%s" device combination is not allowed.
6208	The TSI bus parameter for voice and data terminal devices is not necessary during login.
6209	The connection type EXTERNAL is not allowed for logged off or logged on station.
6210	Sign off" of station is not possible since bus extensions are still existing.

6211	"%s" device combination is not supported for the system type HICOM300E V1.0.
6212	The Cordless E registration is not necessary.
6213	Another mobile station is already existing on the mobile station port with the main extension "%s".
6214	The mobile station port with the main extension "%s" cannot be extended with the "%s" device combination.
6215	The PEN of the mobile station port cannot be found in system "%s" with the main extension "%s".
6216	The bus parameter TSI for voice terminal device is not necessary for mobile station.
6217	The standard key layout of the mobile station cannot be found.
6218	For dummy Cordless E stations, only the parameters "Remarks" and "Billing Account" can be changed.
6219	MSN numbers are set up at the S0 bus or OPTISET connection. Only some of the parameters can be modified.
6220	The PEN of the "logged in" card station can"t be changed.
6221	The "%s" device combination is not allowed for the card station.
6222	For voice device OPTISET, only the TSI number "1" is allowed.
6223	TSI number "%s" of the data terminal device is not allowed.
6224	TSI number "%s" of the fax terminal device is not allowed.
6225	The bus or OPTISET connection with the main extension "%s" is prepared for a relocation or network-wide configuration.
6226	The HICOM300E V2.0 system type does not support the "%s" device combination.
6227	It is not possible to enter the features.
6228	It is not possible to set the KN and KNOVR features simultaneously.
6229	It is not possible to set the HOT and OFHKRC features simultaneously.
6230	It is not possible to set the HOT feature and an ACD feature simultaneously.
6231	It is only possible to set one ANI feature.
6232	It is not possible to set the KN, RKNOVR, KNOVR, VC, HOT, EOVR and OFHKRC features with the "%s" device combination.
6233	It is not possible to set the OFHKRC feature with the "%s" device combination.
6234	It is not possible to set the ANI features with the "%s" device combination.
6235	The display name must not be longer than 30 characters.
6236	The network-wide configuration is not authorized with the "%s" device combination.
6237	"%s" device combination is not suitable for the network-wide configuration "%s" for a station not connected at the bus.
6238	The bus parameter TSI for voice terminal devices is not required for the extended connection.

6239	You must enter the extended connection protocol type.
6240	The "%s" device combination does not need the Protocol and Option parameters for the extended connection.
6241	It is not possible to enter the protocol type for the extended connection.
6242	It is not possible to move special station.
6243	The station "%s" is a member of a Hunt Group. The "%s" device combination is not authorized.
6244	The station "%s" is a member of a Pickup Group. The "%s" device combination is not authorized.
6245	"%s" device combination and connection type "%s" are not suitable for the network-wide configuration "%s" for a station connected to the bus.
6246	It is not possible to set the CW authorization in the first class of service "%s" for the "%s" device combination.
6247	It is not possible to set the CW authorization in the second class of service "%s" for the "%s" device combination.
6248	You must enter the PEN to the networking configuration "%s".
6249	The "%s" device combination does not match the connection type "%s".
6250	The PEN adjacent to the remote SLMO module is already occupied.
6251	The PEN "%s" does not match the "%s" device combination.
6252	It is not possible to set the ANI features for the "%s" device combination and the PEN "%s".
6253	The PEN "%s" does not match the network-wide configuration "%s".
6254	It is not possible to enter the Class of Service Changeover Group.
6255	It is not possible to enter the Basic Speed Dial List.
6256	It is not possible to enter the Speed Dial List.
6257	It is not possible to enter the individual call number.
6258	It is not possible to modify the PEN and connection type for the network-wide configuration "%s".
6259	You must enter the extension number of the destination data for the NT extended connection or HTB system manager.
6260	It is not possible to modify the remote subscriber data with the network-wide configuration "%s".
6261	It is not possible to modify the extended connection data with the network configuration "%s".
6262	It is not possible to modify the extended connection destination data with the network-wide configuration "%s".
6263	The network-wide configuration parameters are not authorized.
6264	You must enter the destination data number and system for the NT extended connection or HTB system manager.
6265	The destination data extension number "%s" and call number "%s" specified for the network-wide configuration do not match.

6266	The destination data system "%s" for the extended connection must differ from the user"s own system.
6267	The destination data for subscriber "%s" on system "%s" does not match the extended connection
6268	The subscriber "%s" does not exist on the system "%s".
6269	The subscriber "%s" does not exist.
6270	The system specified for the destination data and the system specified for the remote subscriber must correspond for the extended connection.
6271	You must enter the extension on the local system and the destination data number for the extended connection
6272	The extension "%s" at own system and the logical subscriber number must differ for the extended connection.
6273	The "%s" device combination does not match the network-wide configuration "%s" for a subscriber connected at the bus.
6274	The extension number "%s" of the destination data does not match the already existing extended connection.
6275	It is not possible to determine the network-wide configuration.
6276	You must enter all of the data for the extended connection.
6277	The extension number "%s" of the extended call does not match the already existing destination data subscriber.
6278	The remote subscriber already exists on the system.
6279	The extension number "%s" and the extension at own system "%s" for the extended connection do not match.
6280	The extension "%s" of the destination data must differ from the extension of the remote subscriber for the extended connection.
6281	The system "%s" specified for the remote subscriber must differ from the user"s own system.
6282	The extension "%s" on the user"s own system does not match the already existing extended connection.
6283	The remote subscriber "%s" is not a suitable extended connection.
6284	The system "%s" specified for the destination data does not match the already existing subscriber specified for the destination data.
6285	The extension "%s" specified for the destination data does not match the already existing subscriber specified for the destination data.
6286	The extended connection is incorrect.
6287	The destination data for the extended connection is incorrect.
6288	No extended connection matches the remote subscriber "%s" on the system "%s".
6289	The extension number "%s" does not match the already existing destination data subscriber.

6290	The extension "%s" of the destination data does not match the already existing extended connection.
6291	You must enter all of the data for the remote subscriber.
6292	You must enter all of the data for the extended connection and the remote subscriber.
6293	The PEN "%s" for the remote subscriber does not match the already existing remote subscriber.
6294	The system "%s" of the remote station is not suitable for the remote station currently available.
6295	The extension "%s" for the remote subscriber does not match the already existing remote subscriber.
6296	Subscriber "%s" at system "%s" is not a suitable remote subscriber.
6297	The PEN "%s" does not match the subscriber "%s" at the system "%s".
6298	The extension "%s" does not match the subscriber at the system "%s" and PEN "%s".
6299	The PEN "%s" for the remote subscriber at the system "%s" does not match the network-wide configuration "%s".
6300	Remote station PEN "%s" does not match the PEN of the existing partner data station.
6301	The protocol type of the functional voice terminal in "%s" device combination must be entered.
6302	The protocol type of the functional TTX terminal in "%s" device combination must be entered.
6303	The protocol type of the functional VTX terminal in "%s" device combination must be entered.
6304	The protocol type of the functional S0/S2 point-to-point connection in "%s" device combination must be entered
6305	The "%s" device combination does not require the Log, TSI and Option parameters for Language.
6306	The "%s" device combination does not require the Log and Option parameters for Language.
6307	The "%s" device combination w i t h o u t TTX terminal does not require the Log, TSI and Option parameters for Language.
6308	The "%s" device combination w i t h o u t VTX terminal does not require the Log, TSI and Option parameters for Language.
6309	The "%s" device combination does not require the Log and Option parameters for S0PP/S2PP.
6310	The "%s" device combination does not require a value in the "%s" field.
6311	The "%s" device combination does not require a value in the "%s" and "%s" fields.
6312	The connection type "%s" requires a value in the "%s" and "%s" fields.
6313	The "%s" device combination does not require a Log and Option parameter.
6314	The "%s" device combination only requires one of the TSI parameters for Language, Fax, DTE, TTX or VTX as a default for the analogue terminal.
6315	The "%s" device combination with analogue DTE device does not require an optional parameter.
6316	The "%s" device combination with analogue FAX terminal does not require an optional parameter.
6317	The "%s" device combination with analogue TTX terminal does not require an Option parameter.

6318	The "%s" device combination with analogue VTX terminal does not require an Option parameter.
6319	At least one functional terminal (fax, DTE, TTX, VTX) in the "%s" device combination requires the protocol type since only one analogue terminal is permissible without a protocol.
6320	At least two analogue functional terminals (fax, DTE, TTX, VTX) in the "%s" device combination require the protocol type since only one analogue terminal is permissible without a protocol.
6321	At least three analogue functional terminals (fax, DTE, TTX, VTX) of the "%s" device combination require the protocol type since only one analogue terminal is permissible without a protocol.
6322	The "%s" device combination does not permit an "%s" parameter.
6323	The "%s" device combination does not permit an "%s" parameter, since a functional terminal already exists (fax , DTE , TTX , VTX).
6324	The "%s" device combination does not permit an "%s" parameter, since there is no subscriber with functional terminal (fax, DTE, TTX, VTX) connected at the bus or OPTISET port.
6325	The properties "%s" and "%s" must not be set in the "%s" device combination.
6326	The properties "%s" and "%s" must not be set simultaneously.
6327	The "%s" device combination cannot be installed at a bus.
6328	No other functional S2PP port can be installed at the "%s" system.
6329	No other fixed call forwarding with the DSS1 protocol can be configured at the "%s" system.
6330	No other ATM port (port switching) can be configured at the "%s" system.
6331	ATM subscribers (port switching) are not permissible.
6332	The functional S0/S2 port with the main extension "%s" cannot be extended with the "%s" device combination.
6333	Field "%s" cannot be modified.
6334	An a/b adapter cannot be configured at the OPTISET port with the main extension "%s".
6335	The TSI number "%s" in the "%s" field is already assigned.
6336	The TSI number "%s" in the "%s" field is not permissible.
6337	The TSI number "%s" in the "%s" field cannot be modified.
6338	The system cannot locate the PEN of the functional S0/S2 port at the system "%s" with the main extension "%s".
6339	The only TSI number permitted for the analogue terminal is "8".
6340	Only two S2PP ports to the external cordless system are permitted at the "%s" system.
6341	The system cannot locate a preset system language at the system "%s".
6342	The STMA card data for the PEN "%s" at the system "%s" has not yet been created.
6343	The value in the "%s" field for the STMA card with the value "%s" in the "%s" field does not fall in the permitted range between "%s" and "%s".

6344	The combination "%s" = "%s" and "%s" = "%s" is already assigned to an ATM subscriber for the STMA card.
6345	The system cannot locate a free VCI value at the STMA card with the value "%s" in the "%s" field.
6346	The value "%s" in the "%s" field is not valid.
6347	The OPTISET port with the main extension "%s" cannot be enhanced with the "%s" device combination. An OPTISET does not require a slave phone, S0 or a/b adapter.
6348	The "%s" device combination with headset is not allowed.
6349	An available matching PEN at "%s" system cannot be determined (on an OPTISET on SLMQ_EXT module).
6350	Fixed call forwarding can only be administered on system "%s".
6351	The station "%s" has a fixed call forwarding with the DSS1 protocol. The "%s" device combination is not authorised.
6352	The station "%s" has a fixed call forwarding with the DSS1 protocol. Action is not executable.
6353	The station "%s" has a fixed call forwarding. Action is not executable.
6354	"%s" device combination can"t be defined on a bus. An OPTISET line requires: a/b adaptor for analogue device w/o protocol; ISDN or data adaptor for DTE or alternatively, control adaptor for API.
6355	The circuit of PEN "%s" does not match the "%s" device combination.
6356	The special station with main extension "%s" cannot be extended.
6357	The "%s" device combination does not require the TSI parameters for TTX or VTX.
6358	The "%s" device combination only requires one of the TSI parameters for Language, Fax or DTE as a default for the analogue terminal.
6359	The station "%s" on system "%s" has more than one fixed call forwarding. Action is not executable.
6360	The station "%s" on system "%s" has more than one display name. Action is not executable.
6361	The station "%s" on system "%s" has a fixed call forwarding with chained single diversion types. Action is not executable.
6362	The "%s" field will be supported from OpenScape 4000 onwards.
6363	The virtual node id "%s" of type "own" does not exist on system "%s".
6364	It cannot locate a virtual node id of type "own" on system "%s".
6365	Values in the "%s" and "%s" fields are only allowed for "%s".
6366	The "%s" field must not be set.
6367	The "%s" device combination and the port equipment number (PEN) "%s" do not require a value in the "%s" field.
6368	The "%s" device combination requires a value in the "%s" and "%s" fields.
6369	It is only possible to set one special facility of the fields "%s", "%s" and "%s".

6370	It is not possible to set the fields "%s", "%s", "%s" and "%s" for the DVC Handler.
6371	It is only possible to set the fields "%s", "%s", "%s" and "%s" for the DVC Handler.
6372	It is not possible to set the fields "%s" and "%s" simultaneously.
6373	The connection type "%s" does not fit to the "%s" module of PEN.
6374	No further COM groups can be created in system "%s" for "%s" device combination.
6375	The "%s" protocol of "%s" device combination does not fit to the "%s" module of PEN.
6376	Functional devices must have the same NI-BRI protocol.
6377	Functional devices with NI-BRI protocol require a basic service of fields "%s", "%s", "%s" or "%s".
6378	The "%s" device combination with data device does not fit to the "%s" module of PEN.
6379	An OPTISET station on SLMQ3 module cannot be extended.
6380	Functional devices without NI-BRI protocol does not require a basic service of fields "%s", "%s", "%s" and "%s".
6381	The "%s" device combination with data device does not permit the "%s" protocol.
6382	PEN for "%s" device combination has to be entered.
6383	The multiple code calling code "%s" does not exist on system "%s".
6384	The fields "%s" and "%s" have to be entered together.
6385	The property "%s" must not be set in the "%s" device combination.
6386	A hotline index is only allowed with "%s" or "%s" feature.
6387	The "%s" or "%s" feature needs a hotline index.
6388	The hotline index "%s" does not exist on system "%s".
6389	No further LIN can be created in system "%s" for "%s" device combination.

10.4.2 Error Texts (German)

NOTICE: Unlike the grouped tables used for the English-language texts in Section 10.4.1, the German-language error texts in this section are listed strictly in increasing numerical order. Furthermore, while the corresponding English and German messages always share the same ID number, there are some messages in one language for which there is not yet a corresponding message in the other language. Both of these issues will be addressed in a future issue of this manual.

Errors for Port (1) [Range: 4350 - 4430]

4350 Die Gerätekombination "%s" wird für den Anlagentyp HICOM300E V3.0 oder V3.1 nicht unterstützt.

4351 An der Anlage "%s" können keine weiteren Rufjournals für die Gerätekombination "%s" eingerichtet werden.

4352 Das Feld "%s" darf nicht gesetzt sein.

4353 An der Anlage "%s" können keine weiteren Applikationen mit Workstation Protokoll (API) mit der Gerätekombination "%s" eingerichtet werden.

4354 Schnittstellenfehler: Die Primärleitung des Key Systems existiert nicht.

4355 Schnittstellenfehler: Die Primärleitung des Key Systems existiert.

4356 Die Standard-Tastenbelegung des Key Systems kann nicht geändert werden.

4357 Der Umzug des Key Systems ist nicht zugelassen.

4358 Das Einrichten oder Löschen des Key Systems ist nur entweder mit dem OPTISET oder mit dem OPTIP500 Endgerät erlaubt.

4359 Der Teilnehmeranschluß "%s" ist als Sekretär vorbereitet.

4360 Key System und Netzweite Konfiguration schließen sich aus.

4361 Die Gerätekombination "%s" kann nicht eingerichtet werden. Das abgemeldete Key System erlaubt kein ComManager.

4362 Die Gerätekombination "%s" kann nicht eingerichtet werden. Das Key System erfordert ein OPTISET Endgerät.

4363 Die Gerätekombination "%s" kann nicht eingerichtet werden. Auf dem oberen Beistellgerät sind den Leitungstasten noch Sekundärleitungen oder die Primärleitung zugeordnet.

4364 Die Lage und die Anschlußart des Key Systems können nicht geändert werden.

4365 An der Anlage "%s" kann kein weiteres Key System eingerichtet werden.

4366 Der Wert "%s" in dem Feld "%s" ist nicht zugelassen, da die Rufoption der Primärleitung nicht gesetzt ist.

4367 Zum Einrichten des Key Systems wird eine Leitungstaste benötigt.

4368 Die Gerätekombination "%s" kann nicht eingerichtet werden. Auf dem Grundgerät sind den oberen Leitungstasten noch Sekundärleitungen oder die Primärleitung zugeordnet.

4369 Die Merkmale "%s", "%s" und "%s" dürfen bei der Gerätekombination "%s" nicht gesetzt sein.

4370 Der Teilnehmeranschluß "%s" ist "%s" in einer Vorzimmeranlage. Er kann nicht als Key System eingerichtet werden.

4371 Die Gerätekombination "%s" kann auf einen Bus nicht eingerichtet werden. Eine OPTISET-Leitung mit DCI700 erfordert einen S0-Adapter.

4372 Der Teilnehmeranschluß "%s" ist oder ist noch Mitglied eines netzweiten Teams. Es können nur einige Felder geändert werden.

4373 Der Teilnehmeranschluß "%s" ist oder ist noch Mitglied eines netzweiten Teams. Die Aktion kann nicht ausgeführt werden.

4374 Die Merkmale "%s" und "%s" dürfen nicht gesetzt sein.

4375 Das Attribut "%s" darf nicht geändert werden.

4376 Die Standard-Tastenbelegung kann nicht geändert werden, da das Merkmal "%s" oder "%s" gesetzt ist.

4377 Teilnehmeranschluss "%s" ist ein Ausweisleser oder ACL-Support-Teilnehmeranschluss für DR (Direktruf) in netzweit reservierter Anrufübernahmegruppe "%s".

4378 Die Gerätekombination, die Anschlußart und die TSI-Parameter können nicht geändert werden, da am OPTISET-Anschluß noch OPTISET-Erweiterungen vorhanden sind.

4379 Der DTB-Zugriff "%s" ist nur möglich, wenn die DTB-Serverkennzahl in der Anlage "%s" bekannt ist.

4380 Mobilteilnehmer mit Datenendgerät vom Typ "%s" werden nicht unterstützt.

4381 Die Parameter "%s", "%s" und "%s" werden erst ab Cordless E Version 2.3 benötigt.

4382 Die feste Anrufumleitung kann nicht aktiviert werden.

4383 Die Kombination der Parameter "%s" = "%s" und "%s" = "%s" ist nicht zugelassen.

4384 Die Gerätekombination "%s" wird für den Anlagentyp OpenScape 4000 V1.0 nicht unterstützt.

4385 Der Umzug eines Teilnehmeranschlusses mit Tastenlayout wird am Assistant nicht unterstützt.

Errors for Switch [Range: 5870 - 5950]

5870 Der Name der VM Server Anlage muß genau 4 Zeichen lang sein.

5871 Typ und Version einer Anlage müssen eingegeben werden.

5872 Die AMO-Sprache der Anlage muß eingegeben werden.

5873 Land, Area Code und Amtsnummer müssen eingegeben werden.

5874 Die Zielnummer muß eingegeben werden.

5875 Für Endgeräteverbindungen "%s", "%s", "%s" und "%s" ein Wert auf dem Feld "%s" muss eingegeben werden.

5876 Die Knotennummer muß eingegeben werden.

5877 Der Name der Domain kann nicht geändert werden.

5878 Die PIN-Längen können nicht verändert werden.

5879 Die Anlage "%s" ist bereits vorhanden.

5880 Die VMS-Server Anlage "%s" ist weder eine HICOM300 V3.2, V3.3, V3.4 oder HICOM300E V1.0, V2.0, V3.0, V3.1 noch ein eigenständiger Voicemail-Server.

5881 Die VMS-Server Anlage "%s" existiert nicht.

5882 Die VMS-Server Anlage "%s" existiert nicht an der Domain "%s".

5883 Die Knotennummer "%s" ist bereits vergeben.

5884 Die Zielnummer "%s" existiert bereits in der Domain "%s".

5885 Die Upload-DELTA Anforderung für PBX-Daten kann nicht mehr zurückgesetzt werden.

5886 Die Upload-DELTA Anforderung für VMS-Daten kann nicht mehr zurückgesetzt werden.

5887 Die Aktion "Sichern" ist nur an einer HICOM300, HICOM300E, ROLM9006 oder OpenScape 4000 bzw. an einem eigenständigen Voicemail-Server möglich.

5888 Die Aktion "Sichern" ist zur Zeit nicht möglich, da an der Anlage ein Upload läuft.

5889 Die Aktion "Sichern" ist zur Zeit nicht möglich, da an der Anlage ein VMS-Upload läuft.

5890 Ein Upload kann nur an einer HICOM300, HICOM300E, ROLM9006 oder OpenScape 4000 durchgeführt werden.

5891 Ein VMS-Upload kann nur direkt an der Anlage "%s" (Anlage mit VMS-Server) durchgeführt werden.

5892 Ein VMS-Upload kann nur an einer HICOM300 V3.2, V3.3, V3.4 oder HICOM300E V1.0, V2.0, V3.0, V3.1 bzw. an einem eigenständigen Voicemail-Server durchgeführt werden.

5893 Die Anlage hat keinen eigenen VM Server. Es darf daher kein Wert eingegeben werden.

5894 Die Zielnummer kann nicht verändert werden.

5895 Ein VMS-Upload kann nur durchgeführt werden, wenn ein VMS-Server eingetragen ist.

5896 Die Upload-DELTA Anforderung für LCR-Daten kann nicht mehr zurückgesetzt werden.

5897 Die Aktion "Sichern" ist zur Zeit nicht möglich, da an der Anlage ein LCR-Upload läuft.

5898 Ein LCR-Upload kann nur an einer HICOM300 V3.4 oder HICOM300E V1.0, V2.0, V3.0, V3.1 durchgeführt werden.

5899 Andere Anlagen im Netz sind bereits mit einer 3-stufigen Knotennummer eingerichtet. Bei "%s" müssen daher die fehlenden Bereichsnummern für die Ebenen 1 bzw. 2 ergänzt werden.

5900 Andere Anlagen im Netz sind bereits mit einer 2-stufigen Knotennummer eingerichtet. Bei "%s" müssen daher die Bereichsnummern für die Ebenen 1 und 2 entsprechend angepaßt werden.

5901 Version "V3.2", "V3.3" oder "V3.4" in Kombination mit Typ "HICOM300RE" oder "HICOM300UE" wird nicht unterstützt.

5902 Andere Anlagen im Netz sind bereits mit einer 1-stufigen Knotennummer eingerichtet. Bei "%s" müssen daher die Bereichsnummern für die Ebenen 1 und 2 auf 0 gesetzt werden.

5903 Die Administrierung der HICOM-Version "%s" ist nicht lizensiert.

5904 Der Typ "%s" der Anlage ist nicht zugelassen.

5905 Die Sprache kann nicht verändert werden.

5906 Die Knotenkennzahl und die DTB Serverkennzahl können nicht verändert werden.

5907 Es ist bereits eine unterstützte Anlage eingerichtet.

5908 Die Aktion "Hochladen" ist zurzeit nicht möglich, da bereits ein Hochladen an der Anlage durchgeführt wird.

5909 Nur Anlage, Beschreibung, Land, Area Code, Amtsrufnummer und die Cordless-Attribute können modifiziert werden.

5910 Der bevorzugte Richtungsindex muss eingegeben werden.

5911 Nur Systemnamen können über System Management modifiziert werden.

5912 Nur die AMO-Sprache kann über System Management modifiziert werden.

5913 Der Richtungsindex der Anlage "%s" in der Sub-Domain "%s", die von allen anderen Sub-Domains erreichbar ist, eingegeben werden.

5914 Ein Richtungsindex darf nur spezifiziert werden, nachdem eine Sub-Domain für Anlage "%s" eingegeben worden ist.

5915 Die Sub-Domain "%s" wurde bereits in andern Anlagen mit Richtungsindex "%s" spezifiziert.

5916 Der Richtungsindex "%s" wird bereits für Sub-Domain "%s" benutzt.

5917 Der Richtungsindex einer Sub-Domain und der bevorzugte Richtungsindex müssen unterschiedlich sein.

5918 Sub-Domain und Richtungsindex einer Sub-Domain können nicht gleichzeitig geändert werden.

5919 Der Richtungsindex muss spezifiziert werden. Dieser wird von der Transitanlage "%s" zum Weiterleiten an die Anlagen der Sub-Domain "%s" benutzt.

5920 Der Wählcode "%s" ist nicht von Typ QUER oder AMT und kann nicht als Trenncode benutzt werden.

5921 Der Wählcode "%s" existiert nicht. Sie müssen ihn zuerst als Wählcodetyp QUER oder AMT einrichten, bevor er als Trenncode benutzt werden kann.

5922 Das Feld 'Trenncode' ist nur gültig, wenn EPNP-Unterstützung im Domain-Bildschirm aktiviert ist.

Errors for Domain [Range: 5951 - 5970]

5951 Die Domain "%s" ist bereits vorhanden.

5952 Es existiert bereits eine Domain mit der Querwahl "%s".

5953 Der Typ des Rufnummernplans kann nicht nach "%s" geändert werden, da der Domain "%s" bereits zwei oder mehr Anlagen zugeordnet sind.

5954 Die Domain "%s" kann nicht gelöscht werden, da ihr noch Anlagen zugeordnet sind.

5955 Da bereits eine Domain ohne Querwahl existiert, muß die Querwahl eingegeben werden.

Errors for Rufnum [Range: 5981 - 6050]

5981 Der Zustand der Rufnummer kann nicht eingegeben werden.

5982 Die Konsistenz der Rufnummer im Domain kann nicht eingegeben werden.

5983 Die Zielnummer kann nicht eingegeben werden.

5984 Die Rufnummer und ihr Typ müssen eingegeben werden.

5985 Für eine Reservierung der Rufnummer muß sowohl die Anlage als auch die Lage eingegeben werden.

5986 Für eine Reservierung der Rufnummer muß sowohl die Anlage als auch die Lage eingegeben werden.

5987 Der Name der Anlage kann nicht eingegeben werden, er wird bei Zuordnung der Rufnummer zu einem Teilnehmeranschluß eingetragen.

5988 Eine Sammelanschluß-Kennzahl darf keine Zielnummer haben.

5989 Zustand und Konsistenz der Rufnummer können nicht geändert werden.

5990 Die Zielnummer kann nicht verändert werden.

5991 Die Rufnummer und ihr Typ können nicht geändert werden.

5992 Domain und Anlage können nicht geändert werden.

5993 Die Rufnummer "%s" kann nicht eingerichtet werden, da sie an der Domain "%s" bereits als Teilnehmer-Rufnummer existiert.

5994 Die Rufnummer "%s" kann nicht eingerichtet werden, da sie an der Domain "%s" bereits als Sammelanschluß-Kennzahl existiert.

5995 Die Teilnehmer-Rufnummer "%s" existiert bereits an der Domain "%s".

5996 Die Teilnehmer-Rufnummer "%s" kann nicht eingerichtet werden, da sie an der Domain "%s" bereits als Sammelanschluß-Kennzahl existiert.

5997 Die Sammelanschluß-Kennzahl "%s" kann nicht eingerichtet werden, da sie an der Domain "%s" bereits als Teilnehmer-Rufnummer existiert.

5998 Die Sammelanschluß-Kennzahl "%s" existiert bereits an der Anlage "%s".

5999 Eine Rufnummer darf nicht Anfang einer anderen Rufnummer sein.

6000 Ein Rufnummer kann nur gelöscht werden, wenn ihr Zustand "frei" oder "unbekannt" ist.

6001 Die Teilnehmer-Rufnummer "%s" existiert nicht an der Domain "%s".

6002 Die Sammelanschluß-Kennzahl "%s" existiert nicht an der Anlage "%s".

6003 Für Kennzahlpunkte des Typs AMT, QUER, QUVPL, RFAMTUA, RFHAUSUA, VAMTUA, STAFFELW, VF Rufnummer sowie für Cordless E Teilnehmer kann keine Reservierung der Rufnummer an einer Anlage bzw. für eine bestimmte Lage eingetragen werden.

6004 Die Rufnummer "%s" und Rufnummern, die mit "%s" beginnen, sind in der Domain "%s" für Cordless E Teilnehmer reserviert.

6005 Die Rufnummer für fiktive Cordless E Teilnehmer kann nicht gelöscht werden.

6006 Die Rufnummer "%s" ist in der Domain "%s" für fiktive Cordless E Teilnehmer reserviert.

Errors for Port (2) [Range: 6051 - 3749]

6051 Die Rufnummer muß eingeben werden.

6052 Der Name der Anlage muß eingegeben werden.

6053 Der Name der Gerätekombination muß eingegeben werden.

6054 Die erste Berechtigungsklasse (COS 1) muß eingegeben werden.

6055 Die zweite Berechtigungsklasse (COS 2) muß eingegeben werden.

6056 Die Lage "%s" paßt nicht zur Gerätekombination "%s", oder sie ist nur für einen am Bus angeschlossenen Teilnehmer zugelassen.

6057 Die Teilnehmerrufnummer "%s" ist nicht konsistent.

6058 In der Domain "%s" ist die Teilnehmerrufnummer "%s" nicht konsistent.

6059 An der Anlage "%s" ist die Teilnehmerrufnummer "%s" nicht konsistent.

6060 Die Standard-Tastenbelegung zur Gerätekombination "%s" muß eingeben werden.

6061 Die Standard-Tastenbelegung zur Gerätekombination "%s" darf nicht eingegeben werden.

6062 Die Teilnehmerrufnummer "%s" ist nicht frei.

6063 In der Domain "%s" ist die Teilnehmerrufnummer "%s" nicht frei.

6064 An der Anlage "%s" ist die Teilnehmerrufnummer "%s" nicht frei.

6065 Gerätekombination "%s" benötigt den Parametersatz für automatisches Downloading.

6066 Der Teilnehmeranschluß "%s" existiert nicht in der Anlage "%s".

6067 Die Kennzeichen für den Mithöraufmerksamkeitston und den Rückrufschutz dürfen nicht eingegeben werden.

6068 Die Bus Parameter: 'Anschlußart', 'Protokoll', 'TSI' und 'Option' werden nur für einen am Bus angeschlossenen Teilnehmer benötigt.

6069 Die Gerätekombination "%s" benötigt keine Anschlußart.

6070 Beim Umzug innerhalb des Domains muß die Rufnummer geändert werden.

6071 Die Least-Cost-Routing-Berechtigungsklassen dürfen nicht eingegeben werden.

6072 An der Anlage "%s" paßt der Tastenbelegungsstandard "%s" nicht zur Gerätekombination "%s". Ãberprüfen Sie das Beistellgerät.

6073 "%s" ist keine Teilnehmerrufnummer.

6074 In der Domain "%s" ist "%s" keine Teilnehmerrufnummer.

6075 An der Anlage "%s" ist "%s" keine Teilnehmerrufnummer.

6076 Die Rufnummer "%s" existiert nicht.

6077 In der Domain "%s" gibt es die Rufnummer "%s" nicht.

6078 An der Anlage "%s" gibt es die Rufnummer "%s" nicht.

6079 Die Gerätekombination "%s" o h n e FAX-Endgerät benötigt nicht die Parameter 'Protokoll', 'TSI' und 'Option'.

6080 Die Domain zur Anlage "%s" kann nicht ermittelt werden.

6081 Eine freie konsistente Teilnehmerrufnummer an der Anlage "%s" kann nicht ermittelt werden.

6082 Eine freie konsistente Teilnehmerrufnummer an der Domain "%s" kann nicht ermittelt werden.

6083 Die Gerätekombination "%s" gibt es nicht.

6084 Die Lage oder die Anschlußart zur Gerätekombination "%s" muß eingegeben werden.

6085 Die Indizes auf Optionstabelle für funktionale Endgeräte dürfen nicht eingegeben werden.

6086 Schnittstellenfehler: Anlage und/oder Rufnummer fehlen!

6087 An der Anlage "%s" können nicht alle PINs übernommen werden.

6088 Die Gerätekombination "%s" o h n e Daten-Endgerät benötigt nicht die Parameter 'Protokoll', 'TSI' und 'Option'.

6089 Der Protokolltyp des funktionalen DEE-Endgerätes der Gerätekombination "%s" muß eingegeben werden.

6090 Der Protokolltyp des funktionalen FAX-Endgerätes der Gerätekombination "%s" muß eingegeben werden.

6091 Die Anlage "%s" wird für den Umzug nicht unterstützt.

6092 Am OPTISET-Anschluß mit der Hauptrufnummer "%s" sind schon zwei OPTISET-Erweiterungen vorhanden.

6093 Die Gerätekombination "%s" wird für den Anlagentyp HICOM300 V3.2 nicht unterstützt.

6094 Die Gerätekombination "%s" ist für Ausweisteilnehmer oder ACL-Supportteilnehmeranschlüsse nicht erlaubt.

6095 Die Gerätekombination "%s" wird für den Anlagentyp HICOM300 V3.3 nicht unterstützt.

6096 Der Gerätetyp zur Gerätekombination "%s" kann nicht ermittelt werden.

6097 Schnittstellenfehler: Der Name der Domain fehlt!

6098 Die Lage "%s" ist nicht frei.

6099 Die Lage "%s" ist in der Domain "%s" nicht frei.

6100 Die Lage "%s" ist an der Anlage "%s" nicht frei.

6101 Es gibt keine Anlage mit der Lage "%s".

6102 In der Domain "%s" gibt es keine Anlage mit der Lage "%s".

6103 An der Anlage "%s" gibt es die Lage "%s" nicht.

6104 Eine freie passende Lage an der Anlage "%s" kann nicht ermittelt werden.

6105 Die Aktion ist wegen nicht behebbarer Inkonsistenz der Technischen Teilnehmerrufnummer "%s" an der Anlage "%s" nicht ausführbar.

6106 In der Domain "%s" gibt es keine Anlage mit der Berechtigungsklasse "%s".

6107 Die TSI-Nummer "%s" des Sprach-Endgerätes kann nicht geändert werden.

6108 Die erste Kurzrufnummernliste "%s" an der Domain "%s" ist nicht vorhanden.

6109 Die zweite Kurzrufnummernliste "%s" an der Domain "%s" ist nicht vorhanden.

6110 An der Anlage "%s" paßt der Tastenbelegungsstandard "%s" nicht zur Gerätekombination "%s".

6111 An der Anlage "%s" kann kein weiterer S0-Bus eingerichtet werden.

6112 Beim Umzug in ein anderes Domain darf die Rufnummer nicht geändert werden.

6113 Der Teilnehmeranschluß "%s" ist nicht für einen Umzug vorbereitet.

6114 Die Gerätekombination "%s" benötigt keine Parameter für Downloading.

6115 Einem Ausweisteilnehmer wird keine Lage zugeordnet.

6116 Teilnehmer können nicht in Ausweisteilnehmer umgewandelt werden.

6117 Beim Umzug an der gleichen Anlage ist die Lage unbedingt zu ändern.

6118 An der Anlage "%s" können keine weiteren Displaynamen vergeben werden.

6119 An der Anlage "%s" können keine weiteren Teilnehmeranschlüsse eingerichtet werden.

6120 An der Anlage "%s" können keine weiteren Teilnehmeranschlüsse mit individuellem Rufnummerngeber eingerichtet werden.

6121 An der Anlage "%s" können keine weiteren Teilnehmeranschlüsse mit der Gerätekombination "%s" eingerichtet werden.

6122 Der Teilnehmeranschluß "%s" ist ein Technischer Teilnehmeranschluß.

6123 Der Teilnehmeranschluß "%s" ist für einen Umzug vorbereitet.

6124 Die Aktion ist wegen nicht behebbarer Inkonsistenz der Teilnehmerrufnummer "%s" in der Domain "%s" nicht ausführbar.

6125 An der Anlage "%s" gibt es die Downloading-Parameterklasse "%s" nicht.

6126 An der Anlage "%s" gibt es den Downloading-Parametersatz "%s" nicht.

6127 Der Umzug ist für Mobilteilnehmer oder Teilnehmer am Bus- oder OPTISET-Anschluß nicht möglich. Es sind noch Buserweiterungen vorhanden.

6128 Die Gerätekombination "%s" o h n e funktionales DEE-Endgerät benötigt nicht die Parameter 'Protokoll' und 'Option'.

6129 Ein Bus- oder OPTISET-Anschluß mit der Hauptrufnummer "%s" ist nicht vorhanden.

6130 Es gibt keine Anlage mit der Berechtigungsklasse "%s".

6131 Am S0-Bus oder OPTISET-Anschluß sind MSN-Rufnummern eingerichtet. Die Aktion kann nicht ausgeführt werden.

6132 An der Anlage "%s" gibt es die Berechtigungsklasse "%s" nicht.

6133 Bei der nicht unterstützten Gerätekombination "%s" oder bei einem Teilnehmer mit ACD-Merkmalen oder bei einem ATM-Teilnehmer (TFA) kann die Aktion nicht ausgeführt werden.

6134 Bei der nicht unterstützten Gerätekombination "%s" oder bei einem Teilnehmer mit ACD-Merkmalen oder bei einem ATM-Teilnehmer (TFA) können nur einige Parameter geändert werden.

6135 Die erste Kurzrufnummernliste und die zweite Kurzrufnummernliste "%s" sind an der Anlage "%s" nicht vorhanden.

6136 Die erste Kurzrufnummernliste "%s" ist nicht vorhanden.

6137 Die erste Kurzrufnummernliste "%s" an der Anlage "%s" ist nicht vorhanden.

6138 Die zweite Kurzrufnummernliste "%s" an der Anlage "%s" ist nicht vorhanden.

6139 Die zweite Kurzrufnummernliste "%s" ist nicht vorhanden.

6140 Die Gerätekombination "%s" kann an einem Bus nicht eingerichtet werden. Ein OPTISET-Anschluß erfordert: a/b-Adapter für analoges Endgerät ohne Protokoll; S0-Adapter für Daten-Endgeräte oder wahlweise Daten-Adapter für funktionale Endgeräte mit Protokoll.

6141 Die Gerätekombination "%s" paßt nicht zur Lage "%s" und Anschlußart "%s".

6142 Der Teilnehmeranschluß kann nicht gelöscht werden, da am Bus-Anschluß mit der Hauptrufnummer "%s" noch Erweiterungen vorhanden sind.

6143 Die Gerätekombination, die Lage und die Anschlußart können nicht geändert werden, da am Bus- oder OPTISET-Anschluß noch Bus- oder OPTISET-Erweiterungen vorhanden sind.

6144 Die Lage "%s" entspricht nicht der Lage am Bus- oder OPTISET-Anschluß mit der Haupt-Rufnummer "%s".

6146 Die Anschlußart "%s" entspricht nicht der Anschlußart am Bus- oder OPTISET-Anschluß mit der Haupt-Rufnummer "%s".

6147 Die Lage des Busanschlusses an Anlage "%s" mit der Hauptrufnummer "%s" kann nicht ermittelt werden.

6148 Der Bustyp am S0-Bus oder am OPTISET-Anschluß mit der Haupt-Rufnummer "%s" kann nicht ermittelt werden.

6149 Die Anschlußart am Bus-, funktionalen S0-/S2- oder OPTISET-Anschluß mit der Hauptrufnummer "%s" kann nicht ermittelt werden.

6150 Die Anschlußart und die Lage können bei einer Bus- oder OPTISET-Erweiterung nicht geändert werden.

6151 Der S0-Bus mit der Hauptrufnummer "%s" kann nicht mit der Gerätekombination "%s" erweitert werden.

6152 Es gibt keine Anlage mit der Downloading-Parameterklasse "%s".

6153 Am Busanschluß mit der Hauptrufnummer "%s" sind schon drei Buserweiterungen vorhanden.

6154 Am S0-Bus können nicht mehr als acht Endgeräte eingerichtet werden.

6155 Die TSI-Nummer "%s" des Daten-Endgerätes kann nicht geändert werden.

6156 Die TSI-Nummer "%s" des Fax-Endgerätes kann nicht geändert werden.

6157 Die TSI-Nummer "%s" des Sprach-Endgerätes ist bereits vergeben.

6158 Die TSI-Nummer "%s" des Daten-Endgerätes ist bereits vergeben.

6159 Die TSI-Nummer "%s" des Fax-Endgerätes ist bereits vergeben.

6160 Es gibt keine Anlage mit dem Downloading-Parametersatz "%s".

6161 Ausweisteilnehmer sind am S0-Bus nicht zugelassen.

6162 Der Umzug von Ausweisteilnehmern ist nicht zugelassen.

6163 An der Anlage "%s" können keine weiteren Direktruftasten oder Namenstasten vergeben werden.

6164 Zur Rufnummer "%s" existiert schon ein VMS-Postfach.

6165 Der Teilnehmeranschluß "%s" ist Mitglied in einer Nachtvariante. Eine Administration an der Anlage "%s" ist erforderlich.

6166 Der Teilnehmeranschluß "%s" ist noch assoziiert. Eine Administration an der Anlage "%s" ist erforderlich.

6167 Der Teilnehmeranschluß "%s" wird noch als Analog-Seite eines Modems benutzt. Eine Administration an der Anlage "%s" ist erforderlich.

6168 Der Teilnehmeranschluß "%s" ist noch CLC-fähig. Eine Administration an der Anlage "%s" ist erforderlich.

6169 Der Ausweisteilnehmer kann nicht umgewandelt werden.

6170 Die Rufnummer und die Lage des für den Umzug vorbereiteten Teilnehmeranschlusses können nicht geändert werden.

6171 Die Rufnummer und die Lage des Technischen Teilnehmeranschlusses können nicht geändert werden.

6172 Die Gerätekombination "%s" wird für den Anlagentyp HICOM300 V3.4 nicht unterstützt.

6173 Die Least-Cost-Routing-Berechtigungsklassen für Daten dürfen nicht eingegeben werden.

6174 Die Least-Cost-Routing-Berechtigungsklassen müssen eingegeben werden.

6175 Die Gerätekombination "%s" ist nur für einen am Bus angeschlossenen Teilnehmer zugelassen.

6176 Die Anlage und die Domain können nicht geändert werden.

6177 Die Rufnummer kann nicht geändert werden.

6178 Die Lage des OPTISET-Anschlusses an Anlage "%s" mit der Hauptrufnummer "%s" kann nicht ermittelt werden.

6179 Der OPTISET-Anschluß mit der Hauptrufnummer "%s" kann nicht mit der Gerätekombination "%s" erweitert werden.

6180 Am OPTISET-Anschluß können nicht mehr als drei Endgeräte eingerichtet werden.

6181 Nur zur Anzeige vorgesehene Felder können nicht verändert werden.

6182 Am OPTISET-Anschluß mit der Haupt-Rufnummer "%s" kann kein Phone-Adapter eingerichtet werden.

6183 Am OPTISET-Anschluß mit der Haupt-Rufnummer "%s" kann kein S0-Adapter eingerichtet werden.

6184 Abmelden eines Teilnehmers oder Anmelden eines Ausweisteilnehmers ist nicht zugelassen.

6185 Anmelden eines Ausweisteilnehmers ist nicht zugelassen.

6186 Teilnehmeranschluss, Ausweisteilnehmer oder ACL-Supportteilnehmeranschluss kann nicht geändert werden.

6187 Umzug abgemeldeter Teilnehmeranschlüsse, angemeldeter Ausweisteilnehmer- oder ACL-Supportteilnehmeranschlüsse ist nicht erlaubt.

6188 Abgemeldete Teilnehmer oder Ausweisteilnehmer sind am Bus- oder OPTISET-Anschluß nicht zugelassen.

6189 Abgemeldete Teilnehmer können am Bus nur in Betrieb genommen werden.

6190 Die Gerätekombination "%s" ist für ab- oder angemeldete Teilnehmer nicht zugelassen (Keine Datenendgeräte).

6191 Einem abgemeldeten Teilnehmer wird keine Lage zugeordnet.

6192 Ein Technischer Teilnehmeranschluß kann nicht abgemeldet werden.

6193 Ein für den Umzug vorbereiteter Teilnehmeranschluß kann nicht abgemeldet werden.

6194 Die Lage "%s" ist nicht abgemeldet.

6195 Die Lage "%s" ist in der Domain "%s" nicht abgemeldet.

6196 Die Lage "%s" ist an der Anlage "%s" nicht abgemeldet.

6197 Eine abgemeldete passende Lage an der Anlage "%s" kann nicht ermittelt werden.

6198 Die Gerätekombination "%s" ist für angemeldete Ausweisteilnehmer- oder ACL-Supportteilnehmeranschlüsse nicht erlaubt.

6199 Einarbeitung von HICOM-Daten läuft noch - bitte später erneut versuchen oder nochmal auswählen

6200 Ziel der Anrufumleitung kann nicht der Teilnehmeranschluss selbst sein

6201 Die Haupt-Rufnummer kann nicht geändert werden.

6202 An der Anlage "%s" können keine weiteren Anrufumleitungen eingerichtet werden.

6203 Die Anrufumleitung darf nicht eingegeben werden.

6204 Der OPTISET-Anschluß mit der Hauptrufnummer "%s" und einem Adapter-Steckplatz kann nicht mit der Gerätekombination "%s" erweitert werden.

6205 Bei einer nicht unterstützten Gerätekombination "%s", abgemeldete Teilnehmeranschlüsse, Ausweisteilnehmer oder ACL-Supportteilnehmeranschlüsse kann nicht in Betrieb genommen werden.

6206 Abmelden eines Teilnehmers mit nicht unterstützter Gerätekombination "%s" ist nicht zugelassen.

6207 Der Teilnehmeranschluß "%s" ist Mitglied in einer Vorzimmeranlage. Die Gerätekombination "%s" ist nicht zugelassen.

6208 Der Bus-Parameter TSI für Sprach- und Datenendgeräte wird beim Anmelden nicht benötigt.

6209 Die Anschlußart EXTERNAL ist für ab- oder angemeldete Teilnehmer nicht zugelassen.

6210 Abmelden des Teilnehmers ist nicht möglich, da noch Buserweiterungen vorhanden sind.

6211 Die Gerätekombination "%s" wird für den Anlagentyp HICOM300E V1.0 nicht unterstützt.

6212 Die Cordless E Registrierung wird nicht benötigt.

6213 Am Mobilteilnehmeranschluß mit der Hauptrufnummer "%s" ist schon ein weiterer Mobilteilnehmer vorhanden.

6214 Der Mobilteilnehmeranschluß mit der Hauptrufnummer "%s" kann nicht mit der Gerätekombination "%s" erweitert werden.

6215 Die Lage des Mobilteilnehmeranschlusses an Anlage "%s" mit der Hauptrufnummer "%s" kann nicht ermittelt werden.

6216 Der Bus-Parameter TSI für Sprachendgeräte wird für Mobilteilnehmer nicht benötigt.

6217 Die Standard-Tastenbelegung des Mobilteilnehmers kann nicht ermittelt werden.

6218 Bei fiktiven Cordless E Teilnehmern können nur die Parameter: Bemerkung und Kostenstelle geändert werden.

6219 Am S0-Bus oder OPTISET-Anschluß sind MSN-Rufnummern eingerichtet. Es können nur einige Parameter geändert werden.

6220 Die Lage des angemeldeten Ausweisteilnehmers kann nicht geändert werden.

6221 Die Gerätekombination "%s" ist für den Ausweisteilnehmer nicht zugelassen.

6222 Für das Sprachendgerät OPTISET ist nur die TSI-Nummer "1" zugelassen.

6223 Die TSI-Nummer "%s" des Datenendgerätes ist nicht zugelassen.

6224 Die TSI-Nummer "%s" des Faxendgerätes ist nicht zugelassen.

6225 Der Bus- oder OPTISET-Anschluß mit der Hauptrufnummer "%s" ist für den Umzug oder die netzweite Konfiguration vorbereitet.

6226 Die Gerätekombination "%s" wird für den Anlagentyp HICOM300E V2.0 nicht unterstützt.

6227 Die Merkmale dürfen nicht eingegeben werden.

6228 Die Merkmale AKL und AUFS dürfen nicht gleichzeitig gesetzt sein.

6229 Die Merkmale HOT und ROE dürfen nicht gleichzeitig gesetzt sein.

6230 Das Attribut HOT und ein ACD-Merkmal dürfen nicht gleichzeitig gesetzt sein.

6231 Nur ein ANI-Attribut darf gesetzt sein.

6232 Die Merkmale AKL, AUFAKS, AUFS, DIRANS, HOT, NATR und ROE dürfen bei der Gerätekombination "%s" nicht gesetzt sein.

6233 Das Attribut ROE darf bei der Gerätekombination "%s" nicht gesetzt sein.

6234 Die ANI-Merkmale dürfen bei der Gerätekombination "%s" nicht gesetzt sein.

6235 Der Displayname darf nicht länger als 30 Zeichen sein.

6236 Die Gerätekombination "%s" erlaubt keine netzweite Konfiguration.

6237 Die Gerätekombination "%s" paßt nicht zur netzweiten Konfiguration "%s" für einen nicht am Bus angeschlossenen Teilnehmer.

6238 Der Bus-Parameter TSI für Sprachendgeräte wird für die Verlängerungsverbindung nicht benötigt.

6239 Der Protokolltyp der Verlängerungsverbindung muß eingegeben werden.

6240 Die Gerätekombination "%s" benötigt die Parameter 'Protokoll' und 'Option' für die Verlängerungsverbindung nicht.

6241 Der Protokolltyp für die Verlängerungsverbindung darf nicht eingegeben werden.

6242 Der Umzug der netzweiten Konfiguration ist nicht zugelassen.

6243 Der Teilnehmeranschluß "%s" ist Mitglied in einem Sammelanschluß. Die Gerätekombination "%s" ist nicht zugelassen.

6244 Der Teilnehmeranschluß "%s" ist Mitglied in einer Anrufübernahmegruppe. Die Gerätekombination "%s" ist nicht zugelassen.

6245 Die Gerätekombination "%s" und die Anschlußart "%s" passen nicht zur netzweiten Konfiguration "%s" für einen am Bus angeschlossenen Teilnehmer.

6246 Die Berechtigung CW in der ersten Berechtigungsklasse "%s" darf für die Gerätekombination "%s" nicht gesetzt sein.

6247 Die Berechtigung CW in der zweiten Berechtigungsklasse "%s" darf für die Gerätekombination "%s" nicht gesetzt sein.

6248 Die Lage zur netzweiten Konfiguration "%s" muß eingegeben werden.

6249 Die Gerätekombination "%s" paßt nicht zur Anschlußart "%s".

6250 Die benachbarte Lage der SLMO Baugruppe vom Typ Remote ist bereits belegt.

6251 Die Lage "%s" paßt nicht zur Gerätekombination "%s".

6252 Die ANI-Merkmale dürfen bei der Gerätekombination "%s" und der Lage "%s" nicht gesetzt sein.

6253 Die Lage "%s" paßt nicht zur netzweiten Konfiguration "%s".

6254 Die Berechtigungsumschaltegruppe darf nicht eingegeben werden.

6255 Die Basiskurzrufnummernliste darf nicht eingegeben werden.

6256 Die Kurzrufnummernliste darf nicht eingegeben werden.

6257 Der individuelle Rufnummerngeber darf nicht eingegeben werden.

6258 Die Lage und die Anschlußart können bei der netzweiten Konfiguration "%s" nicht geändert werden.

6259 Die erweiterte Rufnummer der Partnerdaten für die Verlängerungsverbindung muß eingegeben werden.

6260 Die abgesetzten Teilnehmerdaten können bei der netzweiten Konfiguration "%s" nicht geändert werden.

6261 Die Verlängerungsverbindungsdaten können bei der netzweiten Konfiguration "%s" nicht geändert werden.

6262 Die Partnerdaten der Verlängerungsverbindung können bei der netzweiten Konfiguration "%s" nicht geändert werden.

6263 Die Parameter der netzweiten Konfiguration sind nicht zugelassen.

6264 Die Rufnummer und die Anlage der Partnerdaten für die Verlängerungsverbindung müssen eingegeben werden.

6265 Für die Verlängerungsverbindung passen die erweiterte Rufnummer "%s" und die Rufnummer "%s" der Partnerdaten nicht zusammen.

6266 Für die Verlängerungsverbindung muß sich die Anlage "%s" der Partnerdaten von der eigenen Anlage unterscheiden.

6267 Die Partnerdaten des Teilnehmers "%s" an der Anlage "%s" passen nicht zur Verlängerungsverbindung.

6268 Den Teilnehmer "%s" gibt es an der Anlage "%s" nicht.

6269 Den Teilnehmer "%s" gibt es nicht.

6270 Für die Verlängerungsverbindung müssen die Anlage der Partnerdaten und die Anlage des abgesetzten Teilnehmers übereinstimmen.

6271 Für die Verlängerungsverbindung müssen die Rufnummer an eigener Anlage und die erweiterte Rufnummer der Partnerdaten eingegeben werden.

6272 Für die Verlängerungsverbindung muß sich die Rufnummer "%s" an eigener Anlage von der Rufnummer des logischen Teilnehmers unterscheiden.

6273 Die Gerätekombination "%s" paßt nicht zur netzweiten Konfiguration "%s" für einen am Bus angeschlossen Teilnehmer.

6274 Die erweiterte Rufnummer "%s" der Partnerdaten paßt nicht zur bereits vorhandenen Verlängerungsverbindung.

6275 Die netzweite Konfiguration kann nicht ermittelt werden.

6276 Alle Daten der Verlängerungsverbindung müssen eingegeben werden.

6277 Die erweiterte Rufnummer "%s" der Verlängerungsverbindung paßt nicht zum bereits vorhandenen Teilnehmer der Partnerdaten.

6278 Den abgesetzten Teilnehmer gibt es bereits.

6279 Für die Verlängerungsverbindung passen die erweiterte Rufnummer "%s" und die Rufnummer "%s" an eigener Anlage nicht zusammen.

6280 Für die Verlängerungsverbindung muß sich die Rufnummer "%s" der Partnerdaten von der Rufnummer des abgesetzten Teilnehmers unterscheiden.

6281 Die Anlage "%s" des abgesetzten Teilnehmers muß sich von der eigenen Anlage unterscheiden.

6282 Die Rufnummer "%s" an eigener Anlage paßt nicht zur bereits vorhandenen Verlängerungsverbindung.

6283 Der Teilnehmer "%s" ist keine passende Verlängerungsverbindung.

6284 Die Anlage "%s" der Partnerdaten paßt nicht zum bereits vorhandenen Teilnehmer der Partnerdaten.

6285 Die Rufnummer "%s" der Partnerdaten paßt nicht zum bereits vorhandenen Teilnehmer der Partnerdaten.

6286 Die Verlängerungsverbindung paßt nicht.

6287 Die Partnerdaten der Verlängerungsverbindung passen nicht.

6288 Der Teilnehmer "%s" an der Anlage "%s" ist keine passende Verlängerungsverbindung.

6289 Die erweiterte Rufnummer "%s" paßt nicht zur bereits vorhandenen Verlängerungsverbindung.

6290 Die Rufnummer "%s" der Partnerdaten paßt nicht zur bereits vorhandenen Verlängerungsverbindung.

6291 Alle Daten des abgesetzten Teilnehmers müssen eingegeben werden.

6292 Alle Daten der Verlängerungsverbindung und des abgesetzten Teilnehmers müssen eingegeben werden.

6293 Die Lage "%s" des abgesetzten Teilnehmers paßt nicht zum bereits vorhandenen abgesetzten Teilnehmer.

6294 Die Anlage "%s" des abgesetzten Teilnehmers paßt nicht zum bereits vorhandenen abgesetzten Teilnehmer.

6295 Die Rufnummer "%s" des abgesetzten Teilnehmers paßt nicht zum bereits vorhandenen abgesetzten Teilnehmer.

6296 Der Teilnehmer "%s" an der Anlage "%s" ist kein passender abgesetzter Teilnehmer.

6297 Die Lage "%s" paßt nicht zum Teilnehmer "%s" an der Anlage "%s".

6298 Die Rufnummer "%s" paßt nicht zum Teilnehmer an der Anlage "%s" und der Lage "%s".

6299 Die Lage "%s" des abgesetzten Teilnehmers an der Anlage "%s" paßt nicht zur Netzweiten Konfiguration "%s".

6300 Die Lage "%s" des abgesetzten Teilnehmers paßt nicht zur Lage des bereits vorhandenen Teilnehmers der Partnerdaten.

6301 Der Protokolltyp des funktionalen Sprachendgerätes der Gerätekombination "%s" muß eingegeben werden.

6302 Der Protokolltyp des funktionalen TTX-Endgerätes der Gerätekombination "%s" muß eingegeben werden.

6303 Der Protokolltyp des funktionalen BTX-Endgerätes der Gerätekombination "%s" muß eingegeben werden.

6304 Der Protokolltyp der funktionalen S0/S2-Punkt-Punkt-Verbindung der Gerätekombination "%s" muß eingegeben werden.

6305 Die Gerätekombination "%s" benötigt nicht die Parameter 'Protokoll', 'TSI' und 'Option' für Sprache.

6306 Die Gerätekombination "%s" benötigt nicht die Parameter 'Protokoll' und 'Option' für Sprache.

6307 Die Gerätekombination "%s" o h n e TTX-Endgerät benötigt nicht die Parameter 'Protokoll', 'TSI' und 'Option' für Sprache.

6308 Die Gerätekombination "%s" o h n e BTX-Endgerät benötigt nicht die Parameter 'Protokoll', 'TSI' und 'Option' für Sprache.

6309 Die Gerätekombination "%s" benötigt nicht die Parameter 'Protokoll' und 'Option' für S0PP/S2PP.

6310 Die Gerätekombination "%s" benötigt keinen Wert im Feld "%s".

6311 Die Gerätekombination "%s" benötigt keinen Wert in den Feldern "%s" und "%s".

6312 Die Anschlußart "%s" benötigt einen Wert in den Feldern "%s" und "%s".

6313 Die Gerätekombination "%s" benötigt keine 'Protokoll' und 'Option' Parameter.

6314 Die Gerätekombination "%s" benötigt nur einen der TSI Parameter für Sprache, Fax , DEE ,TTX oder BTX als Standard des analogen Endgerätes.

6315 Die Gerätekombination "%s" mit analogem DEE-Endgerät benötigt den Parameter 'Option' nicht.

6316 Die Gerätekombination "%s" mit analogem FAX-Endgerät benötigt den Parameter 'Option' nicht.

6317 Die Gerätekombination "%s" mit analogem TTX-Endgerät benötigt den Parameter 'Option' nicht.

6318 Die Gerätekombination "%s" mit analogem BTX-Endgerät benötigt den Parameter 'Option' nicht.

6319 Mindestens ein funktionales Endgerät (Fax, DEE, TTX, BTX) der Gerätekombination "%s" benötigt den Protokolltyp, da nur ein analoges Endgerät ohne Protokoll zugelassen ist.

6320 Mindestens zwei funktionale Endgeräte (Fax, DEE, TTX, BTX) der Gerätekombination "%s" benötigen den Protokolltyp, da nur ein analoges Endgerät ohne Protokoll zugelassen ist.

6321 Mindestens drei funktionale Endgeräte (Fax, DEE, TTX, BTX) der Gerätekombination "%s" benötigen den Protokolltyp, da nur ein analoges Endgerät ohne Protokoll zugelassen ist.

6322 Die Gerätekombination "%s" erlaubt keinen "%s" Parameter.

6323 Die Gerätekombination "%s" erlaubt keinen "%s" Parameter, da bereits ein funktionales Endgerät (Fax , DEE , TTX , BTX) vorliegt.

6324 Die Gerätekombination "%s" erlaubt keinen "%s" Parameter, da am Bus- oder OPTISET-Anschluß kein Teilnehmer mit funktionalem Endgerät (Fax , DEE , TTX , BTX) angeschlossen ist.

6325 Die Leistungsmerkmale "%s" und "%s" dürfen in der Gerätekombination "%s" nicht gesetzt sein.

6326 Die Leistungsmerkmale "%s" und "%s" dürfen nicht gleichzeitig gesetzt sein.

6327 Die Gerätekombination "%s" kann an einem Bus nicht eingerichtet werden.

6328 An der Anlage "%s" kann kein weiterer funktionaler S2PP-Anschluß eingerichtet werden.

6329 An der Anlage "%s" kann keine weitere feste Anrufumleitung mit DSS1-Protokoll eingerichtet werden.

6330 An der Anlage "%s" kann kein weiterer ATM-Anschluss (port switching) eingerichtet werden.

6331 ATM-Teilnehmer (port Switching) sind nicht zugelassen.

6332 Der funktionale S0-/S2-Anschluß mit der Hauptrufnummer "%s" kann nicht mit der Gerätekombination "%s" erweitert werden.

6333 Das Feld "%s" kann nicht geändert werden.

6334 Am OPTISET-Anschluß mit der Hauptrufnummer "%s" kann kein a/b-Adapter eingerichtet werden.

6335 Die TSI-Nummer "%s" im Feld "%s" ist bereits vergeben.

6336 Die TSI-Nummer "%s" im Feld "%s" ist nicht zugelassen.

6337 Die TSI-Nummer "%s" im Feld "%s" kann nicht geändert werden.

6338 Die Lage des funktionalen S0-/S2-Anschlusses an der Anlage "%s" mit der Hauptrufnummer "%s" kann nicht ermittelt werden.

6339 Für das analoge Endgerät ist nur die TSI-Nummer "8" zugelassen.

6340 An der Anlage "%s" sind nur zwei S2PP-Anschlüsse zum externen Cordless-System zugelassen.

6341 Eine voreingestellte Systemsprache an der Anlage "%s" kann nicht ermittelt werden.

6342 Die STMA Baugruppendaten für die Lage "%s" an der Anlage "%s" sind noch nicht eingerichtet.

6343 Für die STMA Baugruppe mit dem Wert "%s" im Feld "%s" ist der Wert im Feld "%s" nicht im zulässigen Bereich von "%s" bis "%s".

6344 Für die STMA Baugruppe ist die Kombination "%s" = "%s" und "%s" = "%s" bereits einem ATM-Teilnehmer zugeordnet.

6345 Ein freier VCI-Wert an der STMA Baugruppe mit dem Wert "%s" im Feld "%s" kann nicht ermittelt werden.

6346 Der Wert "%s" im Feld "%s" ist nicht zugelassen.

6347 Der OPTISET-Anschluß mit der Hauptrufnummer "%s" kann nicht mit der Gerätekombination "%s" erweitert werden. Ein Phone-, S0- oder a/b-Adapter ist bei einer OPTISET-Gerätekombination nicht zugelassen.

6348 Die Gerätekombination "%s" mit Kopfsprechgarnitur ist nicht zugelassen.

6349 Eine freie passende Lage an der Anlage "%s" kann nicht ermittelt werden (Beim Optiset an der Baugruppe SLMQ_EXT).

6350 Die feste Anrufumleitung kann nur an der Anlage "%s" administriert werden.

6351 Der Teilnehmeranschluss "%s" weist eine feste Anrufumleitung mit dem DSS1-Protokoll auf. Die Gerätekombination "%s" ist nicht zulässig.

6352 Hauptrufnummer "%s" im System "%s" weist eine feste Anrufumleitung mit dem DSS1-Protokoll auf. Die Aktion kann nicht ausgeführt werden.

6353 Der Teilnehmeranschluss "%s" weist eine feste Anrufumleitung auf. Die Aktion kann nicht ausgeführt werden.

6354 Entweder kann die Gerätekombination "%s" nicht auf einem Bus definiert werden oder das Protokoll für Fax oder DEE fehlt. Bitte stellen Sie außerdem sicher, dass die Gerätekombination alle erforderlichen Adapter enthält.

6355 Der Satz von Lage "%s" passt nicht zu der Gerätekombination "%s".

6356 Der Sondereinrichtungs-Teilnehmer mit Hauptrufnummer "%s" kann nicht erweitert werden.

6357 Die Gerätekombination "%s" benötigt keine TSI-Parameter für TTX oder VTX.

6358 Die Gerätekombination "%s" benötigt nur einen der TSI-Parameter für Sprache, Fax oder DEE als Standard für das analoge Endgerät.

6359 Der Teilnehmeranschluss "%s" an Anlage "%s" weist mehr als eine feste Anrufumleitung auf. Die Aktion kann nicht ausgeführt werden.

6360 Der Teilnehmeranschluss "%s" an Anlage "%s" weist mehr als einen Displaynamen auf. Die Aktion kann nicht ausgeführt werden.

6361 Der Teilnehmeranschluss "%s" an Anlage "%s" weist eine feste Anrufumleitung mit DSS1-Protokoll. Die Aktion kann nicht ausgeführt werden.

6362 Das Feld "%s" wird erst ab OpenScape/OpenScape 4000 unterstützt.

6363 Die virtuelle Knoten-ID "%s" des Typs "eigen" existiert nicht an Anlage "%s".

6364 Eine virtuelle Knoten-ID "%s" des Typs "eigen" kann an Anlage "%s" nicht ermittelt werden.

6365 Die Kombination der Werte in den Feldern "%s" und "%s" ist nicht zulässig.

6366 Das Feld "%s" darf nicht gesetzt werden.

6367 Die Gerätekombination "%s" und die Teilnehmerlage (PEN) "%s" erfordern keinen Wert im Feld "%s".

6368 Die Gerätekombination "%s" erfordert einen Wert in den Feldern "%s" und "%s".

6369 Es kann nur eine einzige Sondereinrichtung der Felder "%s", "%s" und "%s" gesetzt werden.

6370 Es ist nicht möglich, die Felder "%s", "%s", "%s" und "%s" für den DVC-Handler zu setzen.

6371 Es ist nur möglich, die Felder "%s", "%s", "%s" und "%s" für den DVC-Handler zu setzen.

6372 Es ist nicht möglich, die Felder "%s" und "%s" gleichzeitig zu setzen.

6373 Der Verbindungstyp "%s" passt nicht zum Modul "%s" der Lage.

6374 Es können keine weiteren COM-Gruppen an Anlage "%s" für die Gerätekombination "%s" eingerichtet werden.

6375 Das Protokoll "%s" der Gerätekombination "%s" passt nicht zum Modul "%s" der Lage.

6376 Funktionsfähige Geräte müssen das gleiche NI-BRI-Protokoll aufweisen.

6377 Funktionsfähige Geräte mit NI-BRI-Protokoll erfordern einen Basisdienst für die Felder "%s", "%s", "%s" oder "%s".

6378 Die Gerätekombination "%s" mit Datenendgerät passt nicht zum Modul "%s" der Lage.

6379 Ein OPTISET-Teilnehmeranschluss an Modul SLMQ3 kann nicht erweitert werden.

6380 Funktionsfähige Geräte ohne NI-BRI-Protokoll erfordern keinen Basisdienst für die Felder "%s", "%s", "%s" oder "%s".

6381 Die Gerätekombination "%s" mit Datenendgerät lässt das Protokoll "%s" nicht zu.

6382 Die Lage für die Gerätekombination "%s" muss eingegeben werden.

6383 Die Kennzahl für Mehrfach-Personensuche "%s" ist an der Anlage "%s" nicht vorhanden.

6384 Die Felder "%s" und "%s" müssen zusammen eingegeben werden.

6385 Das Leistungsmerkmal "%s" darf mit dieser Gerätekombination "%s" nicht gesetzt sein.

6386 Ein Hotline-Index ist nur bei den Leistungsmerkmalen "%s" oder "%s" erlaubt.

6387 Die Leistungsmerkmale "%s" oder "%s" erfordern einen Hotline-Index.

6388 Der Hotline-Index "%s" ist an der Anlage "%s" nicht vorhanden.

6389 Es kann keine weitere LIN an Anlage "%s" für die Gerätekombination "%s" eingerichtet werden.

6390 Nur ein MLPP-Merkmal darf gesetzt sein (außer MLPPS).

6391 Der Teilnehmeranschluss "%s" hat bereits eine feste Anrufumleitung ohne DRS1-Protokoll. Die Gerätekombination "%s" ist nicht zulässig.

6392 Für die Felder "%s" und "%s" muss ein Wert in das Feld "%s" eingegeben werden.

6393 Die SLC-Gastbaugruppe "%s" mit SLMC-Kartennummer "%s" kann nicht für mobile Stationen benutzt werden.

6394 Die Anschlussart des Key Systems kann nicht geändert werden.

6395 Die Sprache "%s" stimmt nicht mit der Systemsprache des Systems "%s" überein.

6396 Das Leistungsmerkmal "%s" benötigt das Leistungsmerkmal "%s" oder "%s".

6397 Das Leistungsmerkmal "%s" benötigt das Leistungsmerkmal "%s".

6398 Die ACD-Leistungsmerkmale "%s" dürfen mit dieser Gerätekombination "%s" nicht gesetzt sein.

6399 Die Leistungsmerkmale "%s" und "%s" dürfen mit dieser Gerätekombination "%s" nicht gesetzt sein.

6400 Es kann keine weitere Optiset E / optiPoint 500-Verbindung über IP (HFA-Gateway) im System "%s" hergestellt werden.

6401 Die Optiset E / optiPoint 500-Verbindung über IP (HFA-Gateway) mit der Hauptnebenstelle "%s" kann nicht vermittelt werden.

6402 Die Optiset E / optiPoint 500-Verbindungn über IP (HFA-Gateway) benötigt keine Datengeräte und Adapter.

6403 Es können nur maximal vier DNIL-Merkmale eingestellt werden.

6404 Die Gerätekombination "%s" kann in der HICOM300 nicht eingerichtet werden. Ein OPTIP500-Gerät mit 19 Tasten erfordert ein Beistellgerät.

6405 Teilnehmeranschluss "%s" ist Mitglied einer Nachtschaltung.

6406 Die COM-Gruppe "%s" existiert nicht.

6407 Keine Lagennummer ist an einem ACL-Supportteilnehmeranschluss zuogeordnet.

6408 Anmelden des Ausweisteilnehmers oder ACL-Supportteilnehmers ist nicht erlaubt.

6409 Die Lagennummer "%s" ist in der Anlage "%s" bereits reserviert.

6410 Während des Ã„nderns der Tastenbelegungsnummer können Rufnummer und Gerätekombination nicht geändert werden.

6411 Die Rufnummer eines Teilnehmers mit ACD-Merkmalen kann nicht geändert werden.

6412 Die Gerätekombination "%s" wird für den Anlagentyp OpenScape 4000 V2.0 nicht unterstützt.

6413 Leitungsspezifische Leistungsmerkmale stehen nur für Key Systeme zur Verfügung.

Errors for PIN [Range: 6540 - 6590]

6540 Der Teilnehmeranschluss mit Domain, Anlage und Rufnummer muss bekannt sein.

6541 Der PIN-Typ muss eingegeben werden.

6542 Die PIN muss bekannt sein.

6543 Der Teilnehmeranschluss mit Domain, Anlage und Rufnummer kann nicht geändert werden.

6544 PIN-Typ, PIN-Klasse und individuelle PIN können nicht geändert werden

6545 Die PIN "%s" ist bereits vergeben an Anlage "%s".

6547 Die PIN "%s" ist an Anlage "%s" nicht vorhanden.

6548 Für den Teilnehmer "%s" an Anlage "%s" kann keine PIN vergeben werden, da sein Endgerät dies nicht zulässt.

6549 Es kann keine neue PIN eingerichtet werden, da das Limit an Anlage "%s" erreicht ist.

6550 Es kann keine freie PIN mehr ermittelt werden.

6551 Die PIN "%s" vom PIN-Typ "%s" muss genau %s Zeichen lang sein.

6552 Die PIN "%s" kann nicht gelöscht werden, da sie für das Postfach des Teilnehmers "%s" verwendet wird.

6553 Der PIN-Typ "%s" ist an Anlage "%s" nicht zugelassen.

6554 Die Position kann bei dem PIN-Typ "%s" nicht eingegeben werden.

6555 Die Cordless E-PIN "%s" ist bereits vergeben.

6556 Der Teilnehmer "%s" ist kein Mobilteilnehmer.

6557 Der Teilnehmer "%s" hat bereits den Berechtigungsumschaltecode "%s" der PIN-Klasse "%s" an Anlage "%s".

6558 Die Cordless E-PIN "%s" ist nicht erlaubt.

6559 Domain und Anlage müssen bekannt sein.

6560 Der PIN-Typ "%s" darf keinem Teilnehmer zugeordnet werden.

6561 Der PIN-Typ oder die PIN-Klasse müssen eingegeben werden.

6562 Der PIN-Typ "%s" zu der PIN-Klasse "%s" ist an Anlage "%s" nicht zugelassen.

6563 Der PIN-Typ zu der PIN-Klasse "%s" an Anlage "%s" kann nicht ermittelt werden.

6564 Die PIN-Klasse zu dem PIN-Typ "%s" an Anlage "%s" kann nicht oder nicht eindeutig ermittelt werden.

6565 Der Längeneintrag kann nicht geändert werden - alle PINs vom PIN-Typ "%s" müssen zunächst gelöscht werden.

6566 Die PIN-Klasse "%s" ist an Anlage "%s" nicht vorhanden.

6567 Die Anlage und die VBZ-Gruppe müssen bekannt sein.

6568 Die VBZ-Gruppe "%s" ist an Anlage "%s" bereits vorhanden.

6569 Die VBZ-Gruppe "%s" ist an Anlage "%s" nicht vorhanden.

6570 Modulo "%s" ist an Anlage "%s" nicht vorhanden.

6571 Die PIN-Länge reicht von min. "%s" bis max. "%s" Ziffern.

6572 Es ist nicht erlaubt, eine PIN-Klasse hinzuzufügen/zu löschen.

6573 Es ist nicht erlaubt, eine PIN-Klasse zu ändern.

6574 COPIN-Typ "%s" ist nicht erlaubt, wenn DEAKTAST = "%s".

6575 Für dieses Attribut ist das erforderliche Format: x-x-x-x-x-x-x-x-x, wobei x eine Ziffer von 0 bis 9 ist.

6576 Es ist nicht erlaubt, ein PIN-Allgemein-Objekt hinzuzufügen/zu löschen.

6577 Es ist nicht erlaubt, ein PIN-Allgemein-Objekt zu ändern.

6578 Es ist nicht erlaubt, ein PIN-Prüfung-Objekt hinzuzufügen/zu löschen.

6579 Es ist nicht erlaubt, ein PIN-Prüfung-Objekt zu ändern.

6580 Es ist nicht erlaubt, ein Firmenkennzahl-Objekt hinzuzufügen/zu ändern/ zu löschen.

6581 Anlage, Domain und Modul können nicht geändert werden.

6582 Die Firmenkennzahl und der Zeiger müssen bekannt sein.

6583 Die VBZ-Gruppe kann nicht geändert werden.

6584 PIN-Klasse kann nicht geändert werden.

6585 PIN Art kann nicht geändert werden.

6586 Aktion "Ã„ndern" ist nicht erlaubt.

6587 PIN-Klasse muss vorhanden sein.

6588 PIN-Typen, die Projektcodes entsprechen, dürfen nicht geändert werden.

6589 Das Feld "%s" muss die gleiche Länge wie der Wert des Feldes "%s" (%s) haben.

Errors for Devconst [Range: 6651 - 6710]

6651 Die Zusatzgeräte 3 und 4 können nicht eingegeben bzw. geändert werden.

6652 Der Name der Gerätekombination muß eingegeben werden.

6653 Die Gerätefamilie und der Gerätetyp müssen eingegeben werden.

6654 Für Ausweisleser, Freisprechen und Display muß ein Wert eingegeben werden.

6655 Da die Gerätekombination noch mindestens einem Teilnehmeranschluß zugeordnet ist, können nur Felder geändert werden, die keine Anlagendaten enthalten.

6656 Die Gerätekombination "%s" existiert bereits.

6657 Die Kombination Gerätefamilie "%s", Gerätetyp "%s", Beistellgerät "%s", Adapter 1 "%s" und Adapter 2 "%s" ist nicht möglich.

6658 Der Gerätetyp "%s" entspricht nicht der Gerätefamilie "%s".

6659 Das Beistellgerät "%s" ist nicht möglich bei der Gerätefamilie "%s".

6660 Das Beistellgerät "%s" ist nicht möglich bei dem Gerätetyp "%s".

6661 Die Gerätekombination "%s" ist nicht vorhanden.

6662 Die Gerätekombination "%s" kann nicht gelöscht werden, da sie noch einem oder mehreren Teilnehmeranschlüssen zugeordnet ist.

6663 nicht unterstützt

6664 Die Gerätefamilie "%s" mit Ausweisleser erlaubt nur maximal drei Beistellgeräte.

6665 Die Adapteranordnung ist nicht zugelassen.

6666 Der Adapter "%s" ist nicht möglich bei der Gerätefamilie "%s".

6667 Der Adapter "%s" ist nicht möglich bei dem Gerätetyp "%s".

6668 Der Parameter "Freisprechen" wird für Geräte des Typs OPTISET nicht administriert.

6669 Für Ausweisleser und Display muß ein Wert eingegeben werden.

6670 Der Gerätetyp "%s" erlaubt nur einen Adapter.

6671 Die Parameter "ComManager", "Headset" und "Recorder" können nur für Optisets eingegeben werden.

6672 Ein Optiset-Endgerät kann nicht gleichzeitig mit Ausweisleser und ComManager eingerichtet werden.

6673 Ein Optiset mit ComManager benötigt vier Beistellgeräte.

6674 Sonstige Zusatzgeräte können nicht eingegeben bzw. geändert werden.

6675 Die Gerätefamilie "%s" mit Ausweisleser erlaubt nur maximal ein Beistellgerät.

6676 Ein optiPoint 500 mit ComManager benötigt 4 Beistellgeräte.

Errors for Class of Service [Range: 6711 - 6770]

6711 Berechtigungen müssen entweder auf ja oder nein gesetzt sein.

6712 Die COS-Nummer muß eingegeben werden.

6713 Die COS-Nummer "%s" ist an der Anlage "%s" bereits vorhanden.

6714 Die maximale Anzahl der für die Anlage "%s" erlaubten Berechtigungsklassen ist bereits erreicht.

6715 Die Berechtigungsklasse "%s" ist nicht vorhanden.

6716 Die Berechtigungsklasse "%s" existiert nicht an der Anlage "%s".

6717 Die Berechtigungsklasse "%s" existiert nicht an der Anlage "%s" der Domain "%s".

6718 Die Berechtigungsklasse "%s" kann nicht gelöscht werden, weil sie noch von Teilnehmeranschlüssen benutzt wird.

6719 Anklopfen und Aufschalten dürfen nicht gleichzeitig gesetzt sein.

6720 Die folgenden Berechtigungen für Sprache dürfen erst ab einer HICOM300 V3.3 gesetzt werden: NATR, NATRS, KWANR, NAVKZ, LOCC, NATC, INATC, LOCD, NATD, TOLLC, LOCCNV, TOLLOP, LCHARG, TESTEQ,

OVCAT, SPECAR, SUBCALL, SUBDB, SUBMSN, MSN, MULTRA, RWBES, DRKEY

6721 Die folgenden Berechtigungen für Daten dürfen erst ab einer HICOM300 V3.3 gesetzt werden: OVCAT, SPECAR, SUBCALL, SUBDB, SUBMSN, MSN, GESP, MULTRA

6722 Die folgenden Berechtigungen für FAX dürfen erst ab einer HICOM300 V3.3 gesetzt werden: OVCAT, SPECAR, SUBCALL, SUBDB, SUBMSN, MSN, GESP, MULTRA

6723 Die folgenden Berechtigungen für Sprache dürfen erst ab einer HICOM300 V3.2 gesetzt werden: RERING, AUTOV, RELCON

6740 Die Anlage, die Domain und die COS-Nummer können nicht verändert werden.

6741 Die COS-Nummer "%s" ist an der Anlage "%s" reserviert für fiktive Cordless E Teilnehmer.

6742 Es wurde ein Wert für ein nicht überschreibbares Feld eingegeben.

Errors for VoiceMail [Range: 6801 - 6860]

6801 Die Rufnummer muß eingeben werden.

6802 Die VM-COS muß eingegeben werden.

6803 Ansage- und Postfachtyp müssen eingegeben werden.

6804 Die Löschklasse muß eingegeben werden.

6805 Die Berechtigung muß eingegeben werden.

6806 Der Teilnehmeranschluß mit Domain, Anlage und Rufnummer kann nicht geändert werden.

6807 Die VM-COS kann nicht geändert werden.

6808 Ansage- und Postfachtyp können nicht geändert werden.

6809 Die Löschklasse kann nicht geändert werden.

6810 Die Berechtigung kann nicht geändert werden.

6811 Die Anlage des Voicemail-Servers kann nicht geändert werden.

6812 Für den Teilnehmeranschluß "%s" existiert an Anlage "%s" bereits ein Postfach.

6813 Der Teilnehmeranschluß "%s" besitzt an der Anlage "%s" keine Briefkastenberechtigung.

6814 Für den Teilnehmeranschluß "%s" an Anlage "%s" ist kein Postfach eingerichtet.

6815 Die PIN muß eingegeben werden.

6816 VM-COS "%s" existiert nicht an der Anlage "%s".

6817 Für den Teilnehmeranschluß "%s" kann kein Postfach eingerichtet werden, da "%s" bereits als Postfachnummer an Anlage "%s" vergeben ist.

6818 Für den Teilnehmeranschluß mit PIN "%s" kann an der Anlage "%s" kein Postfach eingerichtet werden, da schon ein Postfach mit dieser Code-Nummer existiert.

6819 Die Anlage "%s" hat keinen Voicemail-Server.

6820 Die Anzahl der zugewiesenen Rundspruchverteiler übersteigt das Limit von %s Verteilern an der Anlage %s.

6821 Die Anlage (Voicemail-Server), an der das Postfach eingerichtet werden soll, muß bekannt sein.

6822 Das Kennwort darf nur gelöscht werden.

6823 Das Postfach mit Domain, Rufnummer und Anlage (Voicemail-Server) muß bekannt sein.

6824 Die Postfach-Nummer "%s" gibt es an Anlage "%s" (Voicemail-Server) nicht.

6825 Das Postfach kann nicht eingerichtet werden, da das absolute Limit von %s Postfächern an Anlage "%s" erreicht ist.

6826 Das Postfach kann nicht eingerichtet werden, da das absolute Limit von %s Postfächern vom Typ %s an Anlage "%s" erreicht ist.

6827 Das Postfach kann nicht eingerichtet werden, da das absolute Limit von %s Postfächern mit Ansagetyp %s an Anlage "%s" erreicht ist.

Errors for Persdat [Range: 7461 - 7520]

7461 Der Name des Teilnehmers muß eingegeben werden.

7462 Das Land des Teilnehmers muß eingegeben werden.

7463 Der Standort des Gebäudes "%s" muß eingegeben werden.

7464 Es existiert keine Domain mit der Querwahl "%s".

7465 Der Teilnehmer "%s" ist nicht vorhanden.

7466 Die Anlage kann nicht eindeutig ermittelt werden.

7467 Es ist nicht mehr möglich, für den Teilnehmer "%s" an der Anlage "%s" noch eine freie Position zu finden.

7468 Falls dem Teilnehmer eine Organisationseinheit zugeordnet ist, muß die Firma eingegeben werden.

7469 Die Felder Pers-APD-Typ-TBkz und AKZ1 müssen eingegeben werden.

7470 Die Felder Land, Firma und Standort müssen eingegeben werden.

7471 Die Felder Abteilung und Kst-Abtl. müssen eingegeben werden.

7472 Die Felder Leistungsartpräfix und Leistungsarten (mindestens das erste Feld) müssen eingegeben werden.

7473 Das Leistungsartpräfix muß genau 7 Zeichen lang sein.

7474 Die Eingabe "%s" ist nicht korrekt. Es sind nur 4-stellige Zahlen erlaubt.

7475 Der Servername muss eingegeben werden.

7476 Die Postfach ID muss eingegeben werden.

7477 Die PIN muss eingegeben werden.

7478 Der Servername kann nicht geändert werden.

7479 Die Postfach ID kann nicht geändert werden.

7480 Ein Postfach mit ID "%s" existiert bereits.

7481 Ein Postfach mit PIN "%s" existiert bereits.

7482 Sie müssen für den Teilnehmer zuerst ein Postfach einrichten.

7483 Die Personalnummer muss eingegeben werden.

7484 Die Personalnummer muss genau 6 Zeichen lang sein.

7485 Die Personalnummer ist bereits für einen anderen Teilnehmer vergeben.

7486 Wenn ein Postfach eingerichtet ist, müssen Anlage und Rufnummer eingegeben werden.

7487 Die Firma "%s" im Land "%s" ist nicht vorhanden.

7488 Der Standort "%s" ist nicht vorhanden.

7489 Das Gebäude "%s" am Standort "%s" existiert nicht.

7490 Die Organisationseinheit "%s" existiert nicht.

7491 Die Uid muss genau 7 Zeichen lang sein.

7492 Die Uid ist bereits einem anderen Teilnehmer zugeordnet.

7493 Dieser Teilnehmer wird über das User Management administriert. Er kann nicht gelöscht werden.

Errors for Location [Range: 7521 - 7550]

7521 Der Name des Standorts muß eingegeben werden.

7522 Name des Standorts kann nicht geändert werden, da an ihm bereits Teilnehmer eingetragen sind.

7523 Der Standort "%s" ist bereits erfaßt.

7524 Der Standort "%s" ist nicht vorhanden.

7525 Für den Standort "%s" sind noch Teilnehmer eingetragen. Er kann nicht gelöscht werden.

Errors for Building [Range: 7551 - 7580]

7551 Der Name des Gebäudes muß eingegeben werden.

7552 Der Name des Standorts muß eingegeben werden.

7553 Der Name des Standorts kann nicht geändert werden, da an ihm bereits Teilnehmer eingetragen sind.

7554 Der Standort "%s" existiert nicht.

7555 Das Gebäude "%s" am Standort "%s" ist bereits vorhanden.

7556 Das Gebäude "%s" am Standort "%s" existiert nicht.

7557 Für das Gebäude "%s" sind noch Teilnehmer eingetragen. Es kann nicht gelöscht werden.

Errors for Company [Range: 7581 - 7610]

7581 Land und Firma müssen eingegeben werden.

7582 Der Name der Firma kann nicht geändert werden, da für diese Firma bereits Teilnehmer eingetragen sind.

7583 Die Firma "%s" im Land "%s" ist bereits eingetragen.

7584 Die Firma "%s" im Land "%s" ist nicht vorhanden.

Errors for Country [Range: 7611 - 7640]

7611 Ziel- und Ursprungsland müssen eingegeben werden.

7612 Der Area Code muß eingegeben werden.

7613 Ein Area Code für das Ursprungsland "%s" und das Zielland "%s" ist bereits eingetragen.

7614 Das Zielland muß sich vom Ursprungsland unterscheiden.

7615 Ein Area Code für das Ursprungsland "%s" und das Zielland "%s" ist nicht eingerichtet.

Errors for Org [Range: 7641 - 7680]

7641 Mindestens eine der drei Organisationsstufen muß eingegeben werden.

7642 Firma und Land müssen bekannt sein.

7644 Land und Firma können nicht geändert werden.

7645 Die Firma "%s" im Land "%s" existiert nicht.

7649 Die Organisationseinheit "%s" existiert nicht.

7650 Die Organisationseinheit "%s" existiert bereits.

Errors for Button Indiv [Range: 7941 - 7990]

7941 Die individuelle Tastenbelegung wird erst ab HICOM V3.4 unterstützt.

7942 Der Teilnehmeranschluß "%s" ist entweder nicht eingerichtet, oder er kann nicht als Direktrufziel angegeben werden.

7943 Die individuelle Tastenbelegung kann nicht verändert werden, weil weder DR- noch NA-Tasten vorhanden sind.

7944 Die individuelle Tastenbelegung kann für fiktive Cordless E Teilnehmer nicht geändert werden.

7945 An der Anlage "%s" können keine weiteren Direktruftasten oder Namenstasten vergeben werden.

7946 Individuelle Ã„nderungen der Tastenfunktionen sind beim Digitetyp DIGI211 nicht möglich.

7947 Es dürfen maximal 28 %s-Tasten belegt werden.

7948 Folgende Tastenfunktionen für das Key System sind erst ab einer HICOM300E V3.0 erlaubt: ANZLEIT, LEITUNG, EXHALTEN, HALTEN, PRIVAT, RUFAUS, PRIM

7949 Der Teilnehmeranschluß "%s" ist nicht als Key System eingerichtet. Verwenden Sie zum Einrichten das Feld "Key System" im Teilnehmer-Anschluß Fenster.

7950 Der Teilnehmeranschluß "%s" ist als Key System eingerichtet. Verwenden Sie zum Löschen das Feld "Key System" im Teilnehmer-Anschluß Fenster.

7951 Die Rufnummer "%s" der Sekundärleitung muß sich von der eigenen Rufnummer unterscheiden.

7952 Zur Sekundärleitung "%s" muß der Teilnehmeranschluß "%s" als Key System eingerichtet sein. Verwenden Sie zum Einrichten das Feld "Key System" im Teilnehmer-Anschluß Fenster.

7953 Die Sekundärleitung "%s" ist nicht oder nicht als Key System eingerichtet.

7954 Die Sekundärleitung "%s" ist bereits 40 Key System Teilnehmeranschlüssen zugeordnet.

7955 Schnittstellenfehler: Der Teilnehmeranschluß "%s" ist kein Key System.

7956 Schnittstellenfehler: Der Teilnehmeranschluß "%s" ist ein Key System.

7957 Sekundärleitungen dürfen nur einmal Leitungstasten zugeordnet werden.

7958 Die Rufoption der Primärleitung muß gesetzt sein, da die Präferenz ankommender Verbindungen den Wert "%s" hat.

7959 Der Teilnehmeranschluß "%s" ist oder ist noch Mitglied eines netzweiten Teams. Rufumschalte- und Vertretungstasten können nicht administriert werden.

7960 Der Teilnehmeranschluß "%s" ist oder ist noch Mitglied eines netzweiten Teams. Die Direktrufziele, Primär- und Chefleitungen %s können nicht geändert werden.

7961 Der Teilnehmeranschluß "%s" ist oder ist noch Mitglied eines netzweiten Teams. Direktruf- und Leitungstasten können nicht administriert werden.

7962 Der Teilnehmeranschluß "%s" ist oder ist noch Mitglied eines netzweiten Teams. Die Direktrufziele, Primär- und Chefleitungen %s können nicht administriert werden.

7963 Die Sekundärleitungen %s sind oder sind noch Chefleitungen eines netzweiten Teams.

Errors for DISPLAY_NAME [Range: 8181 - 8220]

8181 Der Teilnehmeranschluss "%s" ist an der Anlage "%s" nicht vorhanden.

8182 Der Displayname darf für die Gerätekombination "%s" nicht angegeben werden. Die Aktion kann nicht ausgeführt werden.

8183 Der Displayname darf nicht länger als 30 Zeichen sein.

8184 An der Anlage "%s" können keine weiteren Displaynamen vergeben werden.

8185 Der Teilnehmeranschluss "%s" weist keinen Displaynamen an Position %s auf.

8186 Der Teilnehmeranschluss "%s" weist bereits einen Displaynamen an Position %s auf.

8187 Die maximale Anzahl an Displaynamen wurde überschritten.

8188 An Anlage "%s" können keine weiteren Organisationseinheiten eingerichtet werden.

8189 Das Feld "%s" wird erst ab OpenScape 4000 unterstützt.

8190 Der Teilnehmeranschluss "%s" hat bereits einen Displaynamen. Ab OpenScape 4000 werden mehrere Displaynamen unterstützt.

8191 Der Teilnehmeranschluss "%s" ist für einen Umzug vorbereitet. Die Aktion kann nicht ausgeführt werden.

8192 Der (für den Umzug vorbereitete) Teilnehmeranschluss "%s" hat bereits einen Displaynamen.

8193 Der Teilnehmeranschluss "%s" hat bereits den Displaynamen "%s".

Index

C

Coded values
    ODF 425, 425
COMPIMP 158, 308
CTRYCODES 309

D

DEVCOM 309
DOMAINDATA 310

E

export table GUI
    general information 82

L

LOCIMP 312

O

ODF
    coded values 425, 425
ORGIMP 313, 313

P

PERS 251, 314
PINDATA 316, 316
PORTDATA 317
PORTUPD 329

S

SWITCHDATA 329

T

Trace
    XIE 82

X

XIE
    coded values 425, 425
    COMPIMP 158, 308
    CTRYCODES 309
    DEVCOM 309
    DOMAINDATA 310
    LOCIMP 312
    ORGIMP 313, 313
    PERS 251, 314

PINDATA 316, 316
PORTDATA 317
PORTUPD 329
SWITCHDATA 329

trace 82

© 2024 Mitel Networks Corporation. All Rights Reserved. Mitel and the Mitel logo are trademark(s) of Mitel Networks Corporation. Unify and associated marks are trademarks of Unify Software and Solutions GmbH & Co. KG. All other trademarks herein are the property of their respective owners.