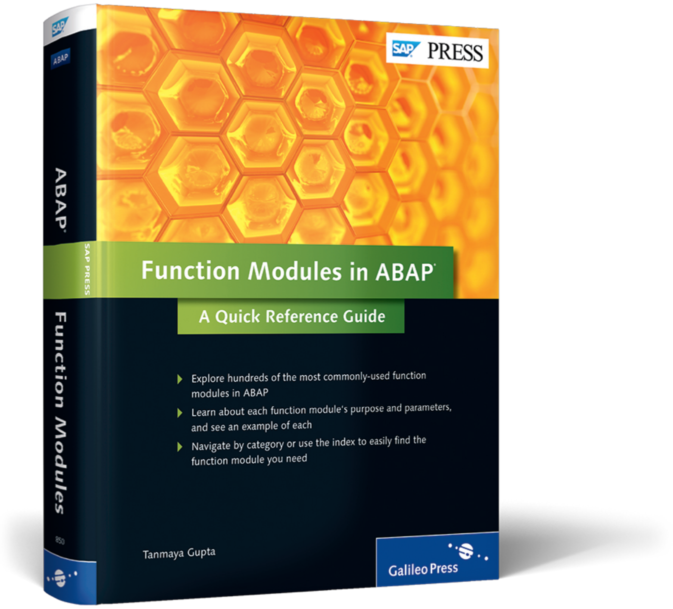
Tanmaya Gupta

Function Modules in ABAP®

A Quick Reference Guide


SAP PRESS
ABAP
ABAP
SAP PRESS
Function Modules
Function Modules in ABAP
A Quick Reference Guide
Explore hundreds of the most commonly-used function modules in ABAP
Learn about each function module's purpose and parameters, and see an example of each
Navigate by category or use the index to easily find the function module you need
Tanmaya Gupta
Galileo Press

Galileo Press®

Bonn • Boston

Contents at a Glance

Part I System 43

Part II ABAP Data Dictionary 83

Part III Conversions 145

Part IV BDC, ALV, Report, and Screens 247

Part V Scripts and Forms 325

Part VI Archive, Number Ranges, Strings, and Text 381

Part VII Frontend Services and Microsoft Office Integration 421

Part VIII ALE, EDI, and IDoc 463

Part IX Users and Authorizations 485

Part X Spools and Background Jobs 505

Part XI Messages and Popups 547

Part XII Transport Management 591

Part XIII Application Log 619

Part XIV BAPI 657

Part XV Miscellaneous 873

Contents

Acknowledgments 33
Introduction 35

Part I: System 43

1 ABAP4_CALL_TRANSACTION 45
2 ABAP_DOCU_DOWNLOAD 46
3 ARFC_GET _TID 47
4 AUTHORITY_CHECK_RFC 47
5 BP_EVENT_RAISE 48
6 CAT_CHECK_RFC_DESTINATION 49
7 DB_COMMIT 50
8 FTP_COMMAND 50
9 FTP_CONNECT 51
10 FTP_DISCONNECT 52
11 HTTP_SCRAMBLE 53
12 OWN_LOGICAL_SYSTEM_GET 54
13 PRETTY_PRINTER 54
14 RFC_CONNECTION_CLOSE 55
15 RFC_READ_TABLE 56
16 RS_DELETE_PROGRAM 58
17 RSLG_WRITE_SYSLOG_ENTRY 60
18 SXPG_CALL_SYSTEM 62
19 SXPG_COMMAND_CHECK 65
20 SXPG_COMMAND_EXECUTE 67
21 SXPG_COMMAND_LIST_GET 69
22 SXPG_DUMMY_COMMAND_CHECK 71
23 TH_DELETE_USER 71
24 TH_ENVIRONMENT 72
25 TH_GET_OWN_WP_NO 73

26 TH_POPUP ... 74
27 TH_REMOTE_TRANSACTION ... 75
28 TH_SERVER_LIST ... 76
29 TH_STOP_WP ... 77
30 TH_USER_INFO ... 78
31 TH_USER_LIST ... 79
32 TRFC_RECEIVER_INFO ... 80
33 USER_EXISTS ... 81

Part II: ABAP Data Dictionary ... 83

34 DDIF_DOMA_GET ... 85
35 DDIF_DTEL_GET ... 86
36 DDIF_FIELDINFO_GET ... 87
37 DDIF_NAMETAB_GET ... 89
38 DDIF_TABL_ACTIVATE ... 92
39 DDIF_TABL_GET ... 92
40 DDIF_TABL_PUT ... 94
41 DDIF_VIEW_GET ... 95
42 DEQUEUE_ALL ... 96
43 DEQUEUE_E_TABLE ... 97
44 ENQUEUE_E_TABLE ... 98
45 F4_DATE ... 101
46 F4IF_DETERMINE_SEARCHHELP ... 103
47 F4IF_FIELD_VALUE_REQUEST ... 104
48 F4IF_GET_SHLP_DESCR ... 107
49 F4IF_INT_TABLE_VALUE_REQUEST ... 108
50 F4IF_SELECT_VALUES ... 111
51 F4IF_SHLP_EXIT_EXAMPLE ... 112
52 F4IF_START_VALUE_REQUEST ... 113
53 F4UT_OPTIMIZE_COLWIDTH ... 114
54 F4UT_PARAMETER_ALLOCATE ... 115

55 F4UT_PARAMETER_RESULTS_PUT 116
56 F4UT_PARAMETER_SORT 117
57 F4UT_PARAMETER_VALUE_GET 118
58 F4UT_RESULTS_MAP 120
59 FLUSH_ENQUEUE 121
60 HELP_DOCULINES_SHOW 121
61 HELP_OBJECT_SHOW 123
62 HELP_OBJECT_SHOW_FOR_FIELD 125
63 HELP_START 127
64 RESET_ENQUEUE 128
65 VIEW_AUTHORITY_CHECK 128
66 VIEW_ENQUEUE 130
67 VIEW_GET_DDIC_INFO 131
68 VIEW_MAINTENANCE 132
69 VIEW_MAINTENANCE_CALL 134
70 VIEW_MAINTENANCE_LOW_LEVEL 137
71 VIEW_MAINTENANCE_SINGLE_ENTRY 140
72 VIEWCLUSTER_MAINTENANCE_CALL 142

Part III: Conversions 145

73 ADD_TIME_TO_DATE 147
74 C14B_ADD_TIME 148
75 CALCULATE_EXCHANGE_RATE 148
76 CF_UT_UNIT_CONVERSION 150
77 CHAR_FLTP_CONVERSION 151
78 COMPUTE_YEARS_BETWEEN_DATES 153
79 CONVERSION_EXIT_ALPHA_INPUT 154
80 CONVERSION_EXIT_ALPHA_OUTPUT 154
81 CONVERSION_EXIT_CUNIT_INPUT 155
82 CONVERSION_EXIT_CUNIT_OUTPUT 155
83 CONVERSION_EXIT_LUNIT_INPUT 156

84 CONVERSION_EXIT_LUNIT_OUTPUT 157
85 CONVERSION_EXIT_GJAHR_INPUT 157
86 CONVERSION_EXIT_GJAHR_OUTPUT 158
87 CONVERSION_EXIT_ISOLA_INPUT 158
88 CONVERSION_EXIT_IDATE_INPUT 159
89 CONVERSION_EXIT_IDATE_OUTPUT 159
90 CONVERSION_EXIT_ISOLA_OUTPUT 160
91 CONVERSION_EXIT_LDATE_OUTPUT 161
92 CONVERSION_EXIT_PERI5_INPUT 161
93 CONVERSION_EXIT_PERI5_OUTPUT 162
94 CONVERSION_EXIT_SDATE_INPUT 162
95 CONVERSION_EXIT_SDATE_OUTPUT 163
96 CONVERSION_FACTOR_GET 163
97 CONVERT_ABAPSPOOLJOB_2_PDF 165
98 CONVERT_DATE_TO_EXTERNAL 167
99 CONVERT_DATE_TO_INTERNAL 168
100 CONVERT_ITF_TO_HTML 168
101 CONVERT_OTF 172
102 CONVERT_OTFSPOOLJOB_2_PDF 174
103 CONVERT_TIME_INPUT 175
104 CONVERT_TO_FOREIGN_CURRENCY 176
105 CONVERT_TO_FRACTION 178
106 CONVERT_TO_LOCAL_CURRENCY 179
107 COUNTRY_CODE_ISO_TO_SAP 180
108 COUNTRY_CODE_SAP_TO_ISO 181
109 CURRENCY_AMOUNT_SAP_TO_BAPI 182
110 CURRENCY_AMOUNT_SAP_TO_IDOC 182
111 CURRENCY_CODE_ISO_TO_SAP 183
112 CURRENCY_CODE_SAP_TO_ISO 184
113 DATE_CHECK_PLAUSIBILITY 185
114 DATE_CHECK_WORKINGDAY 185
115 DATE_COMPUTE_DAY 186

116 DATE_CONV_EXT_TO_INT 186
117 DATE_CONVERT_TO_FACTORYDATE 187
118 DATE_GET_WEEK 188
119 DATE_IN_FUTURE 189
120 DATE_TO_PERIOD_CONVERT 190
121 DAY_ATTRIBUTES_GET 191
122 DAY_IN_WEEK 192
123 DAYS_BETWEEN_TWO_DATES 193
124 DIMENSION_CHECK 195
125 DIMENSION_GET 196
126 DIMENSION_GET_FOR_UNIT 197
127 DURATION_DETERMINE 198
128 END_TIME_DETERMINE 199
129 EVAL_FORMULA 200
130 FACTORYDATE_CONVERT_TO_DATE 201
131 FIMA_DAYS_AND_MONTHS_AND_YEARS 202
132 FIRST_DAY_IN_PERIOD_GET 203
133 FLTP_CHAR_CONVERSION 204
134 GET_CURRENT_YEAR 205
135 HOLIDAY_CHECK_AND_GET_INFO 206
136 HOLIDAY_GET 207
137 HR_HK_DIFF_BT_2_DATES 208
138 HR_SEN_CALE_DAYS_DATE 209
139 HR_SEN_CALE_DAYS_DURATION 209
140 HR_SEN_CRULE_0100_DATE 210
141 HRCM_AMOUNT_TO_STRING_CONVERT 211
142 HRCM_STRING_TO_AMOUNT_CONVERT 211
143 LANGUAGE_CODE_ISO_TO_SAP 212
144 LANGUAGE_CODE_SAP_TO_ISO 213
145 LAST_DAY_IN_PERIOD_GET 213
146 MATERIAL_UNIT_CONVERSION 214
147 MD_CONVERT_MATERIAL_UNIT 216

148 MONTH_NAMES_GET ... 217
149 MONTH_PLUS_DETERMINE ... 217
150 PERIOD_DAY_DETERMINE ... 218
151 READ_EXCHANGE_RATE ... 219
152 RH_GET_DATE_DAYNAME ... 220
153 RKE_TIMESTAMP_CONVERT_INPUT ... 221
154 RKE_TIMESTAMP_CONVERT_OUTPUT ... 222
155 ROUND ... 222
156 RP_CALC_DATE_IN_INTERVAL ... 223
157 RP_LAST_DAY_OF_MONTHS ... 224
158 RS_CONV_EX_2_IN ... 225
159 SCMS_BINARY_TO_XSTRING ... 226
160 SCMS_XSTRING_TO_BINARY ... 227
161 SCP_CODEPAGE_FOR_LANGUAGE ... 228
162 SCP_REPLACE_STRANGE_CHARS ... 228
163 SD_DATETIME_DIFFERENCE ... 230
164 SI_UNIT_GET ... 230
165 START_TIME_DETERMINE ... 231
166 SUBTRACT_TIME_FROM_DATE ... 232
167 SWI_DURATION_DETERMINE ... 233
168 SX_OBJECT_CONVERT_OTF_PDF ... 234
169 SX_OBJECT_CONVERT_OTF_PRT ... 235
170 SX_OBJECT_CONVERT_OTF_RAW ... 236
171 TIME_CHECK_PLAUSIBILITY ... 237
172 UNIT_CONVERSION_SIMPLE ... 237
173 UNIT_CONVERSION_WITH_FACTOR ... 238
174 UNIT_CORRESPONDENCE_CHECK ... 239
175 UNIT_GET ... 240
176 UNIT_OF_MEASURE_ISO_TO_SAP ... 241
177 UNIT_OF_MEASURE_SAP_TO_ISO ... 241
178 UNIT_OF_MEASUREMENT_HELP ... 242
179 WDKAL_DATE_ADD_FKDAYS ... 243

180 WEEK_GET_FIRST_DAY 244
181 WEEKDAY_GET 244
182 WLB3_GET_NUMBER_OF_WORKDAYS 245

Part IV: BDC, ALV, Report, and Screens 247

183 BDC_CLOSE_GROUP 249
184 BDC_INSERT 249
185 BDC_OPEN_GROUP 250
186 COMPLEX_SELECTIONS_DIALOG 252
187 DP_CONTROL_ASSIGN_TABLE 253
188 DP_CREATE 255
189 DP_GET_FIELDS_FROM_TABLE 256
190 DYNP_GET_STEPL 257
191 DYNP_VALUES_READ 257
192 DYNP_VALUES_UPDATE 259
193 FREE_SELECTIONS_DIALOG 260
194 FREE_SELECTIONS_INIT 263
195 FREE_SELECTIONS_RANGE_2_EX 266
196 FREE_SELECTIONS_RANGE_2_WHERE 267
197 LIST_FREE_MEMORY 267
198 LIST_FROM_MEMORY 267
199 LIST_TO_ASCI 268
200 LVC_FIELDCATALOG_MERGE 269
201 LVC_TRANSFER_FROM_SLIS 269
202 LVC_VARIANT_DEFAULT_GET 270
203 LVC_VARIANT_EXISTENCE_CHECK 271
204 LVC_VARIANT_F4 272
205 REUSE_ALV_BLOCK_LIST_APPEND 272
206 REUSE_ALV_BLOCK_LIST_DISPLAY 274
207 REUSE_ALV_BLOCK_LIST_INIT 274
208 REUSE_ALV_COMMENTARY_WRITE 275

209 REUSE_ALV_FIELDCATALOG_MERGE 276
210 REUSE_ALV_GRID_DISPLAY 277
211 REUSE_ALV_GRID_LAYOUT_INFO_GET 282
212 REUSE_ALV_GRID_LAYOUT_INFO_SET 283
213 REUSE_ALV_HIERSEQ_LIST_DISPLAY 283
214 REUSE_ALV_LIST_DISPLAY 286
215 REUSE_ALV_LIST_LAYOUT_INFO_GET 286
216 REUSE_ALV_LIST_LAYOUT_INFO_SET 289
217 REUSE_ALV_POPUP_TO_SELECT 289
218 REUSE_ALV_VARIANT_DEFAULT_GET 291
219 REUSE_ALV_VARIANT_EXISTENCE 292
220 REUSE_ALV_VARIANT_F4 293
221 REUSE_ALV_VARIANT_SELECT 294
222 RPY_DYNPRO_READ 296
223 RPY_TRANSACTION_READ 297
224 RS_COVERPAGE_SELECTIONS 298
225 RS_CREATE_VARIANT 299
226 RS_EXTERNAL_SELSCREEN_STATUS 300
227 RS_FUNCTION_POOL_CONTENTS 301
228 RS_REFRESH_FROM_SELECTOPTIONS 301
229 RS_REPORTSELECTIONS_INFO 302
230 RS_SELECTIONSCREEN_READ 303
231 RS_SET_SELSCREEN_STATUS 303
232 RS_SELECTIONSCREEN_UPDATE 304
233 RS_SUBMIT_INFO 305
234 RS_SUPPORT_SELECTIONS 306
235 RS_TOOL_ACCESS 306
236 RS_TREE_CONSTRUCT 308
237 RS_TREE_LIST_DISPLAY 309
238 RS_VARIANT_CATALOG 311
239 RS_VARIANT_CHANGE 313
240 RS_VARIANT_CONTENTS 314

241 RS_VARIANT_DELETE 315
242 RS_VARIANT_EXISTS 316
243 RS_VARIANT_VALUES_TECH_DATA 317
244 RZL_SUBMIT 318
245 SAPGUI_PROGRESS_INDICATOR 318
246 SCROLLING_IN_TABLE 319
247 SELECT_OPTIONS_RESTRICT 321
248 VIEW_SET_PF_STATUS 322
249 VRM_DELETE_VALUES 322
250 VRM_SET_VALUES 323
251 WRITE_LIST 324
252 WWW_LIST_TO_HTML 324

Part V: Scripts and Forms 325

253 CHECK_TEXT_AUTHORITY 327
254 CHECK_TEXT_ID 327
255 CHECK_TEXT_LANGUAGE 328
256 CHECK_TEXT_NAME 329
257 CHECK_TEXT_OBJECT 329
258 CLOSE_FORM 330
259 COMMIT_TEXT 330
260 CONTROL_FORM 332
261 CONVERT_OTF_MEMORY 333
262 COPY_TEXTS 334
263 CREATE_TEXT 335
264 DELETE_TEXT 336
265 DELETE_TEXT_FROM_CATALOG 336
266 EDIT_TEXT 338
267 EDIT_TEXT_INLINE 340
268 END_FORM 343
269 EXCHANGE_ITF 343

270 EXPORT_TEXT 344
271 FORMAT_TEXTLINES 346
272 FP_FUNCTION_MODULE_NAME 347
273 FP_GET_LAST_ADS_ERRSTR 347
274 FP_JOB_CLOSE 348
275 FP_JOB_OPEN 348
276 IMPORT_TEXT 349
277 INIT_TEXT 350
278 OPEN_FORM 352
279 PRINT_TEXT 355
280 PRINT_TEXT_ITF 357
281 READ_FORM_ELEMENTS 358
282 READ_FORM_LINES 358
283 READ_TEXT 359
284 READ_TEXT_INLINE 361
285 REFER_TEXT 363
286 RENAME_TEXT 364
287 SAVE_TEXT 365
288 SELECT_TEXT 367
289 SSF_FUNCTION_MODULE_NAME 369
290 SSF_GET_DEVICE_TYPE 370
291 SSF_READ_ERRORS 371
292 START_FORM 371
293 TEXT_CONTROL_REPLACE 372
294 TEXT_INCLUDE_REPLACE 373
295 TEXT_SYMBOL_REPLACE 374
296 TEXT_SYMBOL_SETVALUE 376
297 WRITE_FORM 377
298 WRITE_FORM_LINES 379

Part VI: Archive, Number Ranges, Strings, and Text 381
299 ARCHIVE_CLOSE_FILE 383
300 ARCHIVE_GET_CUSTOMIZING_DATA 383
301 ARCHIVE_GET_INFORMATION 384
302 ARCHIVE_GET_NEXT_OBJECT 386
303 ARCHIVE_GET_NEXT_RECORD 387
304 ARCHIVE_GET_STATISTICS 389
305 ARCHIVE_GET_TABLE 390
306 ARCHIVE_GIVE_STATISTICS 392
307 ARCHIVE_NEW_OBJECT 393
308 ARCHIVE_OPEN_FOR_DELETE 393
309 ARCHIVE_OPEN_FOR_MOVE 395
310 ARCHIVE_OPEN_FOR_READ 397
311 ARCHIVE_OPEN_FOR_WRITE 398
312 ARCHIVE_PUT_RECORD 400
313 ARCHIVE_PUT_TABLE 401
314 ARCHIVE_READ_OBJECT 402
315 ARCHIVE_SAVE_OBJECT 404
316 ARCHIVE_WRITE_STATISTICS 405
317 AS_API_READ 406
318 NUMBER_CHECK 406
319 NUMBER_GET_INFO 408
320 NUMBER_GET_NEXT 409
321 NUMBER_RANGE_DEQUEUE 410
322 NUMBER_RANGE_ENQUEUE 411
323 NUMBER_RANGE_INTERVAL_UPDATE 411
324 NUMBER_RANGE_UPDATE_CLOSE 413
325 RKD_WORD_WRAP 413
326 SPELL_AMOUNT 414
327 STRING_CONCATENATE 415
328 STRING_REVERSE 416

329 STRING_UPPER_LOWER_CASE 417
330 SWA_STRING_SPLIT 418
331 TEXT_SPLIT 418

Part VII: Frontend Services and Microsoft Office Integration 421

332 ALSM_EXCEL_TO_INTERNAL_TABLE 423
333 EXCEL_OLE_STANDARD_DAT 423
334 EXECUTE_WINWORD 425
335 FILE_GET_NAME 425
336 FILENAME_GET 428
337 GUI_DOWNLOAD 428
338 GUI_HAS_ACTIVEX 433
339 GUI_IS_AVAILABLE 434
340 GUI_IS_ITS 434
341 GUI_UPLOAD 434
342 KCD_EXCEL_OLE_TO_INT_CONVERT 437
343 KD_GET_FILENAME_ON_F4 438
344 MS_EXCEL_OLE_STANDARD_DAT 439
345 RH_START_EXCEL_WITH_DATA 440
346 RS_SEND_MAIL_FOR_SPOOLLIST 442
347 SAP_CONVERT_TO_XLS_FORMAT 443
348 SO_ATTACHMENT_INSERT_API1 443
349 SO_ATTACHMENT_READ_API1 445
350 SO_DLI_INSERT_API1 446
351 SO_DLI_READ_API1 447
352 SO_DOCUMENT_INSERT_API1 449
353 SO_DOCUMENT_READ_API1 450
354 SO_DOCUMENT_SEND_API1 452
355 SO_NEW_DOCUMENT_ATT_SEND_API1 454
356 SO_NEW_DOCUMENT_SEND_API1 456

357 SO_OLD_DOCUMENT_SEND_API1 458
358 SO_USER_READ_API1 459
359 XXL_FULL_API 460
360 XXL_SIMPLE_API 462

Part VIII: ALE, EDI, and IDoc 463

361 ALE_ASYNC_BAPI_GET_RECEIVER 465
362 ALE_BAPI_GET_FILTEROBJECTS 466
363 ALE_BAPI_GET_UNIQUE_RECEIVER 466
364 ALE_IDOCS_CREATE 467
365 ALE_SYNC_BAPI_GET_RECEIVER 468
366 CHANGE_POINTERS_READ 469
367 CHANGE_POINTERS_STATUS_WRITE 470
368 DEQUEUE_ES_EDIDOCE 471
369 EDI_CHANGE_CONTROL_RECORD 471
370 EDI_CHANGE_DATA_SEGMENTS 472
371 EDI_DOCUMENT_CLOSE_EDIT 473
372 EDI_DOCUMENT_CLOSE_PROCESS 473
373 EDI_DOCUMENT_DEQUEUE_LATER 474
374 EDI_DOCUMENT_OPEN_FOR_EDIT 475
375 EDI_DOCUMENT_OPEN_FOR_PROCESS 476
376 EDI_DOCUMENT_STATUS_SET 477
377 EDI_STATUS_ADD_BLOCK 478
378 IDOC_INBOUND_ASYNCHRONOUS 479
379 IDOC_INBOUND_SINGLE 479
380 IDOC_INBOUND_WRITE_TO_DB 480
381 IDOC_START_INBOUND 481
382 IDOC_STATUS_WRITE_TO_DATABASE 482
383 MASTER_IDOC_DISTRIBUTE 483

Part IX: Users and Authorizations 485

384 ADDR_DIALOG 487
385 ADDR_DIALOG_PREPARE 488
386 ADDR_GET 489
387 ADDR_GET_COMPLETE 491
388 ADDR_INSERT 492
389 ADDR_MEMORY_CLEAR 493
390 ADDR_MEMORY_SAVE 494
391 ADDR_NUMBER_GET 494
392 ADDRESS_INTO_PRINTFORM 496
393 ADDRESS_SHOW_PRINTFORM 499
394 AUTHORITY_CHECK_DATASET 501
395 AUTHORITY_CHECK_TCODE 502
396 SUSR_USER_AUTH_FOR_OBJ_GET 502

Part X: Spools and Background Jobs 505

397 BP_FIND_JOBS_WITH_PROGRAM 507
398 BP_JOB_DELETE 507
399 BP_JOB_READ 509
400 BP_JOB_SELECT 510
401 BP_JOBLOG_READ 512
402 BP_JOBLOG_SHOW 513
403 BP_START_DATE_EDITOR 514
404 GET_JOB_RUNTIME_INFO 517
405 GET_PRINT_PARAMETERS 518
406 JOB_CLOSE 521
407 JOB_OPEN 525
408 JOB_SUBMIT 526
409 LOAD_PRINT_PARAMETERS 528
410 PRINT_SELECTIONS 529
411 RSPO_DISPLAY_SPOOLJOB 530

412 RSPO_DOWNLOAD_SPOOLJOB ... 531
413 RSPO_FIND_SPOOL_REQUESTS ... 531
414 RSPO_OUTPUT_SPOOL_REQUEST ... 533
415 RSPO_RETURN_ABAP_SPOOLJOB ... 535
416 RSPO_RID_SPOOLREQ_LIST ... 536
417 RSTS_CLOSE ... 536
418 RSTS_GET_ATTRIBUTES ... 537
419 RSTS_OPEN_RLC ... 539
420 RSTS_READ ... 540
421 SET_PRINT_PARAMETERS ... 542
422 SHOW_JOBSTATE ... 544
423 STORE_PRINT_PARAMETERS ... 545

Part XI: Messages and Popups ... 547

424 COPO_POPUP_TO_DISPLAY_TEXTLIST ... 549
425 ERGO_TEXT_SHOW ... 549
426 F4_CLOCK ... 550
427 F4_FILENAME ... 551
428 F4_FILENAME_SERVER ... 552
429 FITRV_CALCULATOR ... 552
430 FORMAT_MESSAGE ... 553
431 MD_POPUP_SHOW_INTERNAL_TABLE ... 554
432 MESSAGES_ACTIVE ... 554
433 MESSAGES_COUNT ... 555
434 MESSAGES_GIVE ... 557
435 MESSAGES_INITIALIZE ... 557
436 MESSAGES_SHOW ... 559
437 MESSAGES_STOP ... 560
438 MESSAGE_STORE ... 561
439 POPUP_CONTINUE_YES_NO ... 563
440 POPUP_DISPLAY_TEXT ... 564

441 POPUP_DISPLAY_TEXT_USER_BUTTON ... 565
442 POPUP_DISPLAY_TEXT_WITH_PARAMS ... 566
443 POPUP_FOR_INTERACTION ... 567
444 POPUP_GET_VALUES ... 569
445 POPUP_GET_VALUES_DB_CHECKED ... 570
446 POPUP_GET_VALUES_SET_MAX_FIELD ... 571
447 POPUP_GET_VALUES_USER_BUTTONS ... 572
448 POPUP_GET_VALUES_USER_CHECKED ... 574
449 POPUP_GET_VALUES_USER_HELP ... 576
450 POPUP_NO_LIST ... 577
451 POPUP_TO_CONFIRM ... 578
452 POPUP_TO_DECIDE_LIST ... 580
453 POPUP_TO_DISPLAY_TEXT ... 581
454 POPUP_TO_INFORM ... 582
455 POPUP_TO_SELECT_MONTH ... 582
456 POPUP_WITH_3_BUTTONS_TO_CHOOSE ... 583
457 POPUP_WITH_TABLE_DISPLAY ... 584
458 RPY_MESSAGE_COMPOSE ... 585
459 SO_EXPRESS_FLAG_SET ... 586
460 TERM_CONTROL_EDIT ... 588
461 TXW_TEXTNOTE_EDIT ... 588
462 WRITE_MESSAGE ... 589

Part XII: Transport Management ... 591

463 TR_APPEND_LOG ... 593
464 TR_APPEND_TO_COMM_OBJS_KEYS ... 594
465 TR_CHECK_TYPE ... 597
466 TR_DEVCLASS_GET ... 600
467 TR_FLUSH_LOG ... 601
468 TR_INSERT_NEW_COMM ... 601
469 TR_OBJECT_INSERT ... 604

470 TR_OBJECT_CHECK 606
471 TR_OBJECTS_CHECK 607
472 TR_OBJECTS_INSERT 608
473 TR_READ_COMM 610
474 TR_SYS_PARAMS 612
475 TR_TADIR_INTERFACE 613
476 TR_TADIR_POPUP_ENTRY_E071 616

Part XIII: Application Log 619

477 APPL_LOG_DELETE 621
478 APPL_LOG_DISPLAY 622
479 APPL_LOG_DISPLAY_INTERN 623
480 APPL_LOG_INIT 624
481 APPL_LOG_INIT_MESSAGES 624
482 APPL_LOG_READ_DB 625
483 APPL_LOG_READ_INTERN 626
484 APPL_LOG_SET_OBJECT 628
485 APPL_LOG_WRITE_DB 629
486 APPL_LOG_WRITE_HEADER 629
487 APPL_LOG_WRITE_LOG_PARAMETERS 630
488 APPL_LOG_WRITE_MESSAGE_PARAMS 631
489 APPL_LOG_WRITE_MESSAGES 632
490 APPL_LOG_WRITE_SINGLE_MESSAGE 633
491 BAL_DB_DELETE 634
492 BAL_DB_LOAD 635
493 BAL_DB_SAVE 636
494 BAL_DB_SEARCH 637
495 BAL_DSP_LOG_DISPLAY 637
496 BAL_DSP_LOG_PRINT 639
497 BAL_DSP_PROFILE_DETLEVEL_GET 640
498 BAL_DSP_PROFILE_NO_TREE_GET 641

499 BAL_DSP_PROFILE_POPUP_GET 641
500 BAL_DSP_PROFILE_SINGLE_LOG_GET 642
501 BAL_DSP_PROFILE_STANDARD_GET 643
502 BAL_GLB_AUTHORIZATION_GET 644
503 BAL_GLB_AUTHORIZATION_RESET 645
504 BAL_GLB_CONFIG_GET 645
505 BAL_GLB_CONFIG_SET 646
506 BAL_GLB_MEMORY_REFRESH 647
507 BAL_GLB_SEARCH_LOG 648
508 BAL_GLB_SEARCH_MSG 648
509 BAL_LOG_CREATE 649
510 BAL_LOG_HDR_READ 650
511 BAL_LOG_MSG_ADD 651
512 BAL_LOG_MSG_ADD_FREE_TEXT 652
513 BAL_LOG_MSG_DELETE 653
514 BAL_LOG_MSG_DELETE_ALL 654
515 BAL_LOG_MSG_READ 654
516 BAL_LOG_REFRESH 655

Part XIV: BAPI 657

517 BAPI_ACC_ACTIVITY_ALLOC_POST 659
518 BAPI_ACC_BILLING_POST 660
519 BAPI_ACC_DOCUMENT_CHECK 662
520 BAPI_ACC_DOCUMENT_POST 665
521 BAPI_ACC_DOCUMENT_REV_POST 667
522 BAPI_ACC_GL_POSTING_CHECK 668
523 BAPI_ACC_GL_POSTING_POST 669
524 BAPI_ACC_GL_POSTING_REV_POST 670
525 BAPI_ACC_INVOICE_RECEIPT_CHECK 671
526 BAPI_ACC_INVOICE_RECEIPT_POST 673
527 BAPI_ACC_MANUAL_ALLOC_POST 675

528 BAPI_ALM_CONF_CREATE 676
529 BAPI_ALM_NOTIF_CREATE 677
530 BAPI_ALM_NOTIF_SAVE 679
531 BAPI_ALM_ORDER_GET_DETAIL 679
532 BAPI_ALM_ORDER_MAINTAIN 681
533 BAPI_AR_ACC_GETKEYDATEBALANCE 683
534 BAPI_AR_ACC_GETOPENITEMS 684
535 BAPI_AR_ACC_GETPERIODBALANCES 685
536 BAPI_ASSET_RETIREMENT_POST 685
537 BAPI_BILLINGDOC_CANCEL1 686
538 BAPI_BILLINGDOC_CREATEMULTIPLE 687
539 BAPI_BUPA_ADDRESS_CHANGE 688
540 BAPI_BUPA_ADDRESS_GETDETAIL 691
541 BAPI_BUPA_ADDRESS_REMOVE 693
542 BAPI_BUPA_CENTRAL_CHANGE 693
543 BAPI_BUPA_CENTRAL_GETDETAIL 696
544 BAPI_BUPA_CREATE_FROM_DATA 698
545 BAPI_CHARACT_CREATE 700
546 BAPI_CHARACT_GETDETAIL 701
547 BAPI_CLASS_CHANGE 702
548 BAPI_CLASS_CREATE 704
549 BAPI_CLASS_GET_CHARACTERISTICS 705
550 BAPI_CLASS_GETDETAIL 706
551 BAPI_CLASS_GETSTATUS 707
552 BAPI_COMPANYCODE_GET_PERIOD 708
553 BAPI_COMPANYCODE_GETLIST 709
554 BAPI_CONTRACT_CHANGE 709
555 BAPI_CONTRACT_CREATE 712
556 BAPI_CONTRACT_GETDETAIL 715
557 BAPI_COPAACTUALS_POSTCOSTDATA 718
558 BAPI_COPAPLANNING_GETDATA 719
559 BAPI_COPAPLANNING_POSTDATA 720

560 BAPI_CURRENCY_CONV_TO_EXTERNAL 721
561 BAPI_CURRENCY_CONV_TO_INTERNAL 721
562 BAPI_CUSTOMER_CHANGEFROMDATA1 722
563 BAPI_CUSTOMER_CREATE 723
564 BAPI_CUSTOMER_GETLIST 724
565 BAPI_CUSTOMERINQUIRY_CHANGE 724
566 BAPI_CUSTOMERQUOTATION_CHANGE 727
567 BAPI_ENTRYSHEET_CREATE 729
568 BAPI_EQUI_CHANGE 730
569 BAPI_EQUI_CREATE 731
570 BAPI_FIXEDASSET_CHANGE 732
571 BAPI_FIXEDASSET_CREATE1 735
572 BAPI_GOODSMVT_CANCEL 738
573 BAPI_GOODSMVT_CREATE 739
574 BAPI_GOODSMVT_GETITEMS 740
575 BAPI_HELPVALUES_GET 741
576 BAPI_HU_CREATE 743
577 BAPI_HU_PACK 744
578 BAPI_HU_UNPACK 744
579 BAPI_IBAN_CHANGE 745
580 BAPI_INB_DELIVERY_CONFIRM_DEC 746
581 BAPI_INSPECTIONPLAN_CREATE 748
582 BAPI_INTERNALORDER_CREATE 750
583 BAPI_MATERIAL_AVAILABILITY 751
584 BAPI_MATERIAL_BOM_GROUP_CREATE 753
585 BAPI_MATERIAL_DELETE 754
586 BAPI_MATERIAL_EDIT 754
587 BAPI_MATERIAL_GET_ALL 755
588 BAPI_MATERIAL_GET_DETAIL 757
589 BAPI_MATERIAL_GETLIST 758
590 BAPI_MATERIAL_SAVEDATA 759
591 BAPI_MATERIAL_SAVEREPLICA 762

592 BAPI_MATERIAL_STOCK_REQ_LIST 765
593 BAPI_NETWORK_COMP_ADD 768
594 BAPI_OBJCL_CHANGE 769
595 BAPI_OBJCL_CREATE 770
596 BAPI_OBJCL_GETCLASSES 771
597 BAPI_OBJCL_GETDETAIL 772
598 BAPI_OBJCL_GETDETAIL_KEY 773
599 BAPI_OBJCL_GETSTATUS 774
600 BAPI_OUTB_DELIVERY_CONFIRM_DEC 775
601 BAPI_OUTB_DELIVERY_CREATE_SLS 778
602 BAPI_OUTB_DELIVERY_CREATE_STO 779
603 BAPI_PLANNEDORDER_CHANGE 781
604 BAPI_PLANNEDORDER_CREATE 782
605 BAPI_PLANNEDORDER_DELETE 783
606 BAPI_PLANNEDORDER_EXIST_CHECK 784
607 BAPI_PLANNEDORDER_GET_DETAIL 785
608 BAPI_PLANNEDORDER_GET_DET_LIST 786
609 BAPI_PO_CHANGE 786
610 BAPI_PO_CREATE1 791
611 BAPI_PO_GETDETAIL1 795
612 BAPI_PO_GETITEMS 798
613 BAPI_PO_RELEASE 799
614 BAPI_PO_RESET_RELEASE 801
615 BAPI_PR_CHANGE 802
616 BAPI_PR_CREATE 804
617 BAPI_PRODORD_CHANGE 807
618 BAPI_PRODORD_CREATE 808
619 BAPI_PRODORDCONF_CREATE_HDR 809
620 BAPI_PRODORDCONF_CREATE_TT 810
621 BAPI_PS_INITIALIZATION 812
622 BAPI_PS_PRECOMMIT 812
623 BAPI_QUOTATION_CREATEFROMDATA2 813

624 BAPI_REQUIREMENT_CREATE 815
625 BAPI_REQUIREMENTS_CHANGE 816
626 BAPI_REQUIREMENTS_CREATE 818
627 BAPI_REQUISITION_CHANGE 819
628 BAPI_REQUISITION_CREATE 820
629 BAPI_REQUISITION_DELETE 822
630 BAPI_REQUISITION_RELEASE 822
631 BAPI_RESERVATION_CHANGE 824
632 BAPI_RESERVATION_CREATE1 825
633 BAPI_RESERVATION_GETDETAIL1 826
634 BAPI_ROUTING_CREATE 827
635 BAPI_SAG_CHANGE 829
636 BAPI_SAG_CREATE 832
637 BAPI_SAG_GETDETAIL 836
638 BAPI_SALESORDER_CHANGE 839
639 BAPI_SALESORDER_CREATEFROMDAT2 841
640 BAPI_SALESORDER_GETLIST 844
641 BAPI_SALESORDER_SIMULATE 845
642 BAPI_SCHEDULE_MAINTAIN 847
643 BAPI_SOURCEDETERMIN_GETSOS 848
644 BAPI_TRANSACTION_COMMIT 849
645 BAPI_TRANSACTION_ROLLBACK 850
646 BAPI_USER_ACTGROUPS_ASSIGN 850
647 BAPI_USER_ACTGROUPS_DELETE 851
648 BAPI_USER_CHANGE 852
649 BAPI_USER_CLONE 854
650 BAPI_USER_CREATE1 857
651 BAPI_USER_DELETE 859
652 BAPI_USER_EXISTENCE_CHECK 860
653 BAPI_USER_GET_DETAIL 860
654 BAPI_USER_GETLIST 862
655 BAPI_USER_LOCK 863

656 BAPI_USER_UNLOCK 864
657 BAPI_USER_PROFILES_ASSIGN 864
658 BAPI_USER_PROFILES_DELETE 865
659 BAPI_USR01DOHR_GETEMPLOYEE 865
660 BAPI_VENDOR_CREATE 866
661 BAPI_VENDOR_DELETE 867
662 BAPI_VENDOR_GETDETAIL 867
663 BAPI_WHSE_STOCK_GET_DETAIL 868
664 BAPI_XBP_JOB_CLOSE 868
665 BAPI_XBP_JOB_OPEN 869
666 BAPI_XBP_JOB_STATUS_GET 870
667 BAPI_BUS2001_SET_STATUS 871

Part XV: Miscellaneous 873

668 CALL_BROWSER 875
669 CHANGEDOCUMENT_CLOSE 875
670 CHANGEDOCUMENT_DELETE 877
671 CHANGEDOCUMENT_DISPLAY 878
672 CHANGEDOCUMENT_MULTIPLE_CASE 879
673 CHANGEDOCUMENT_OPEN 880
674 CHANGEDOCUMENT_PREPARE_TABLES 881
675 CHANGEDOCUMENT_READ 882
676 CHANGEDOCUMENT_READ_HEADERS 884
677 CHANGEDOCUMENT_READ_POSITIONS 886
678 CHANGEDOCUMENT_SINGLE_CASE 888
679 CLOI_PUT_SIGN_IN_FRONT 889
680 CS_BOM_EXPL_EQU_V2 889
681 CS_BOM_EXPL_MAT_V2 892
682 CS_WHERE_USED_MAT 896
683 CSAP_MAT_BOM_READ 897
684 CSEP_MAT_BOM_SELECT 899

685 CSEP_MAT_BOM_SELECT_CHANGE 900
686 CSEP_MAT_BOM_SELECT_WHERE_USED 900
687 DEQUEUE_ESFUNCTION 901
688 ENQUEUE_ESFUNCTION 902
689 ENQUE_SLEEP 904
690 FUNCTION_EXISTS 904
691 FUNCTION_IMPORT_INTERFACE 905
692 FUNCTION_INCLUDE_CONCATENATE 907
693 FUNCTION_INCLUDE_INFO 908
694 FUNCTION_INCLUDE_SPLIT 909
695 G_DECIMAL_PLACES_GET 911
696 GET_COMPONENT_LIST 911
697 GET_FIELDTAB 912
698 GET_GLOBAL_SYMBOLS 913
699 GET_INCLUDETAB 913
700 K_WERKS_OF_BUKRS_FIND 914
701 MATERIAL_BTCI_SELECTION_NEW 914
702 MD_MRP_LIST_API 915
703 MD_STOCK_REQUIREMENTS_LIST_API 917
704 NUMERIC_CHECK 918
705 QF05_RANDOM 919
706 QF05_RANDOM_INTEGER 920
707 RS_GET_ALL_INCLUDES 920
708 RV_ORDER_FLOW_INFORMATION 921
709 RZL_SLEEP 922
710 SAPGUI_SET_FUNCTIONCODE 922
711 SCI_INSPECT_LIST 923
712 STATUS_BUFFER_REFRESH 924
713 STATUS_CHANGE_EXTERN 924
714 STATUS_CHANGE_INTERN 925
715 STATUS_CHECK 926
716 STATUS_DELETE_FROM_BUFFER 927

717 STATUS_MAINTAIN 928
718 STATUS_NUMBER_CONVERSION 929
719 STATUS_OBJECT_CREATE 931
720 STATUS_OBJECT_SWITCH_NUMBER 932
721 STATUS_PRE_READ 933
722 STATUS_READ 933
723 STATUS_TEXT_CONVERSION 934
724 STATUS_TEXT_EDIT 935
725 SWC_ELEMENT_GET 937
726 SWC_ELEMENT_SET 937
727 SWE_EVENT_CREATE 938
728 SWC_TABLE_GET 939
729 SWC_TABLE_SET 940
730 SWO_CREATE 940
731 SWO_INVOKE 941
732 SWO_FREE 942

The Author 943
Index 945

Introduction

Function modules (or FMs, as we'll often refer to them throughout the book) are templates that can be used and re-used within ABAP code in order to perform specific tasks. Instead of writing code to perform a certain functionality every time that functionality is needed, programmers can simply call a function module template within the logic. Then the programmer needs only to fill out the template with the details that are specific to his particular program, and the functionality is executed.

For example, a common functionality required in an ABAP program is the need to convert a value from an internal format (how the backend SAP system stores the value) to an external format (how the user should view the value). If this functionality is required, there is no need for programmers to write a separate program every time they need it; rather, they can simply call a function module and supply the necessary details that control how the value should be converted.

The most efficient programmers are those who have a commanding knowledge of function modules, which can save the time of writing unnecessary custom code. However, with about 7,000 function modules provided and released by SAP, understanding all of these FMs isn't an easy task.

This book is your guide to the most commonly used FMs in direct programming activities. We've also included common FMs that are related to SAP functionality but are used in the background. For example, a user may receive an error, and the error root cause is the FM executed in the background. Additionally, SAP uses FMs in its standard delivered transactions. A user might want to redesign and customize those functionalities with some additional options and might need to use an FM.

During our research for the book, we created, analyzed, and worked on a list of thousands of released FMs, as well as some very common but not released FMs. Based on our research, this book names and describes the most commonly used FMs from that list of thousands, which we identified by searching where-used lists for each FM, as well as extensive help and Google searches. Our goal was to provide an invaluable reference for all those who work in ABAP development.

Note

Although you can use unreleased FMs in your programming, SAP offers no guarantee that it will support these FMs in the future or that these FMs are correct. You can go to a FM's ATTRIBUTE tab via Transaction SE37 to see if the FM is released.

Structure of the Book

Based on the functionality of the FMs we've included, we've structured this book to make it very easy to understand. You can search by FM name in the index or table of contents, as well as by key word for what the FM accomplishes or works with. This book contains 15 parts, each of which are created and based on different functionalities and different areas in the SAP system. You may find that some FMs are part of a larger sequence or process and must be used chronologically in order to complete a task. In these cases, the first FM used in the process will contain a list of or mention the other FMs that are used sequentially.

This book has been structured to be as user- and reader-friendly as possible. Each FM is first explained in general terms and then defined with an example to help you understand when, where, and why you can use that FM. Some of the FMs will also contain figures to show how you can use the FM and what results you can expect from the FM call.

We'll now discuss the different parts of this book.

Part I

Part I focuses on the FMs associated with systems. "System" here refers to SAP ERP or SAP NetWeaver systems. This part deals with more of the Basis-related operations such as calling a new transaction, calling SAP NetWeaver from legacy systems, such as Java or Visual Basic (VB), or calling from an FTP server or a remote location. This part also provides you with the FMs related to IP addresses, user authorizations, login and logoff of the user in the system, calling a transaction in another window, and so on.

Part II

This part focuses on the ABAP Data Dictionary. This is the central repository of ABAP, which describes and manages all of the data definitions used in the ABAP system and the database. ABAP Data Dictionary supports the definition of user-defined types, such as data elements, structures, and table types. These types are used in the ABAP processors and programs. It also defines the structure of database objects, such as tables, views, and indexes. These database objects are automatically created in the underlying database with the definition of the ABAP Data Dictionary when the objects are activated. It also provides editing tools such as search help and lock objects.

Part III

This part includes FMs that are related to conversions of values from one format to another, such as from character to float, from one SI unit to another, from spool to PDF, and so on. FMs related to date, time, unit of measure, round off, and so on are also covered in this part. This part also includes CONVERSION_EXIT FMs, which are used to convert the SAP internal format to the user output format and vice versa.

Part IV

This part includes FMs related to Batch Data Communication (BDC), ABAP List Viewer (ALV), report programming, and screen programming. Report and screen (dialog or module pool) programming are the main crux of the ABAP language. Reports are the executable programs that mainly deal with reporting; that is, fetching the data from the database, processing it, and displaying the results. Results are displayed by ALV. This part defines various FMs such as those for calling an ALV, creating a field catalog for ALV, calling the BDC, reading and displaying the screen values, and so on.

Part V

This part focuses on FMs related to form printing. SAP's first form technology was SAPscript. In release 4.6, SAP introduced Smart Forms, which eased the creation of form output by allowing modeling of logic and output using graphical tools. In release 6.40, SAP introduced SAP Interactive Forms by Adobe, where the graphical editor was much easier to use, provided many more functionalities, and rendered output exactly as designed in the editor (PDF). This part contains the FMs used to call the Smart Forms or Adobe Interactive Forms; reading, saving, and updating the SAPscript texts; and so on.

Part VI

This part contains various functions such as archives, number ranges, texts, and strings that you'll use while developing your programs. You'll use archives to store the old data and information (which isn't in use) in your secondary storage so that you free the current memory for usage while still being able to refer to the information whenever required.

Database records of the commercial objects such as purchase documents, sales documents, addresses, and so on are uniquely identified by a number or an ID. Number ranges assign the next available numbers to these commercial object database records and help provide unique identification. This part also contains FMs for text and string operations such as splitting the text based on some separator, concatenating strings, and so on.

Part VII

This part contains the FMs related to frontend services and Microsoft Office integration. Frontend services contain the FMs related to SAP GUI frontends such as GUI_DOWNLOAD, which downloads the FMs to your computer, and GUI_UPLOAD, which uploads the FMs.

Office integration deals with the FMs related to Microsoft Office features such as opening the file in Word and Excel.

Part VIII

An Intermediate Document (IDoc) is the data container used to transfer data between two systems. Application Link Enabling (ALE) and Electronic Data Interchange (EDI) are used for exchanging business data between different systems. Data is distributed through ALE when you want to communicate from one SAP system to one or more other (mostly SAP) systems. ALE transfers the data in IDoc format and uses the tRFC (transactional RFC) method for data transfer. EDI, on the other hand, is generally used if you want to transfer business application document with the external partner system like vendor or customer. This part focuses on the ALE, EDI, and IDoc FMs; that is, FMs used to send IDoc data from one system to another.

Part IX

This part contains the FMs related to SAP users and their authorizations to execute transactions. Users are maintained in Transaction SU01 (User Master), and data are saved in user-specific tables, whose names generally start with "USR" such as

USR01, USR02, USR03, and so on. This part contains the FMs that are used to set user details such as an address, print the address in Smart Forms, and so on. This part also contains the FM to check whether the user has relevant authorizations to execute any transaction, to execute any ABAP Data Dictionary table or view, or to access any file.

Part X

This part contains the FMs related to spools and jobs. A spool request is the document for which a print function has been selected and not yet been printed. The spool system uses the spool request to store the print data temporarily and access it. An output request is created for the spool request document that outputs the print data of the spool request to a particular output device (e.g., printer).

SAP background processing automates routine tasks and helps you optimize your organization's SAP computing resources. Background processing tells the SAP system to run programs for you. It also lets you move long-running or resource-intensive program runs to times when the system load is low. You can delegate the task of running reports or programs to the system. Background processing isn't tied up with the user interaction and dialog screens. This part covers FMs used for creating a spool request, finding the spool request, opening the job, submitting the job for background processing, closing the job, and so on.

Part XI

This part deals with communication with the user via messages or popups. Message texts (standard or custom) are generally used to provide information, warning, success, or error details to the user.

This part contains the FMs for creating a message, sending and collecting the messages for the message handler, and taking user confirmation in popup screens. Popup screens are used to show the time and date, to display help, and so on.

Part XII

The Transport Management System plays an important role in the development and customizing environment. All of the development is locked under the task of the transport in the development system, which is moved to the quality system and finally to the production system. The Transport Management System administers and controls the new development, and records the version of the changes made for a particular program. This part focuses on FMs used in the Transport Management System.

Part XIII

Application logs are used for logging application events. This process is similar to using a system log where system event information is recorded. The advantage of an application log is that it's a system-wide standardized process and is convenient to analyze. Several application logs can be written at the same time by an application. This part defines the FMs related to application logs.

Part XIV

Business Application Programming Interfaces (BAPIs) define the API methods for SAP business objects. These business object types and their BAPIs are described and stored in the Business Object Repository (BOR). BAPIs are always remote function call (RFC) enabled, so they can also be called from external application systems and other programs. BAPIs are also related to SAP components such as Sales and Distribution (SD), Materials Management (MM), Production Planning (PP), Quality Management (QM), Warehouse Management (WM), and so on. This part contains all of the commonly used BAPIs.

Part XV

This part contains miscellaneous FMs that are important and commonly used but can't be categorized into any of the preceding parts. Many component-specific FMs that aren't BAPIs are covered in this part.

Structure of Each Function Module Entry

Each FM can be defined by its parameters (import, export, changing, tables, and exceptions). This book describes all of these parameters and uses them in the FM call. All of these parameters, excluding exceptions, are associated with a data type. In each FM, we've identified each type of parameter with an icon and heading for easier identification.

[↓] Import

Import parameters are the parameters or values that you pass on to the FM to execute it correctly when you're in an ABAP program. Some of the import parameters are mandatory, and the rest are optional. Some import parameters are defined with the default values; that is, if you don't pass any value, then the default values are automatically passed.

[↑] Export

Export parameters contain the results returned from the FM execution.

[Δ] Changing

Changing parameters indicate those parameters whose value is changed in the FM. These parameters can be considered as import and export parameters; for example, an input value is passed in the FM. The value is used, and a new value after execution of the FM is returned in the same parameter.

▦ Tables

Table parameters contain the internal table that you want to pass in the FM. Table parameters are also like changing parameters, in that some of the values can be changed after the execution of the FM.

[!] Exceptions

Exceptions are the errors or some exemptions that might have occurred during FM execution. They generally return the messages that define what problem has occurred. Exceptions should be handled in the program that is calling the FM, and the message explaining the exception should be displayed to the user.

Part IX
Users and Authorizations

This part of the book contains the function modules (FMs) related to users who are using SAP and their authorizations to execute transactions. Users are maintained in Transaction SU01 (User Master), and data are saved in user-specific tables, whose names generally start with "USR" such as USR01, USR02, USR03 and so on.

This part contains the FMs that are used to set user details, such as an address; printing the address in Smart Forms; and so on. This part also contains the FM to check whether the user has relevant authorizations to execute any transaction, to execute any ABAP Data Dictionary table or view, or to access any file.

384 ADDR_DIALOG

This FM displays the dialog box in which you can maintain the complete address of a user, vendor, third party, and so on. FM ADDR_DIALOG_PREPARE (FM 385) must be called if you want to parameterize the dialog box, including its title, field selection, exclude functions, and so on.

Example: You want to maintain the address of any person that exists in your SAP system. If you want to parameterize the address dialog box; that is, cause only those specific settings to appear in the address dialog box, you can call FM ADDR_DIALOG_PREPARE (FM 385) first. You can call this FM to open the address dialog box screen and maintain the address without calling FM ADDR_DIALOG_PREPARE (FM 385), which opens the address dialog box with default settings. You can maintain addresses in this dialog box.

[↓] Import

CHECK_ADDRESS

If this parameter is set to X, the address contents are checked. The default value is X. Type: SZAD_FIELD-FLAG.

SUPPRESS_TAXJURCODE_CHECK

The TAX JURISDICTION CODE field is suppressed and isn't handled if this parameter is X. The default value is [Space]. Type: SZAD_FIELD-FLAG.

IV_TIME_DEPENDENT_COMM_DATA

This parameter determines the time validity for communication data. The default value is [Space]. Type: AD_TIMEDEP.

[↑] Export

OK_CODE

Passed for the calling program. Type: SY-UCOMM.

▦ Tables

NUMBER_HANDLE_TAB

Transfer table of addresses to be processed. Type: ADDR1_DIA.

VALUES

Return table with maintained address data. Type: ADDR1_DATA.

[!] Exceptions

ADDRESS_NOT_EXIST

Address does not exist or is invalid.

GROUP_NOT_VALID

Address group does not exist or is invalid.

PARAMETER_ERROR

Incorrect parameter values.

INTERNAL_ERROR

Serious internal error (MESSAGE A...).

385 ADDR_DIALOG_PREPARE

This FM sets parameters for address maintenance or filtering conditions for field selection, which reduces the parameters of address dialog screen. This FM should be called before calling the address maintenance dialog (from FM ADDR_DIALOG [FM 384]) to set up application-specific parameters for the address dialog. These address settings are valid until the end of the LUW or until the FM is called again and overwrites them.

Example: If you want to change the title bar of the address dialog screen, exclude/include any pushbutton, or apply some other condition, you can pass these parameters in this FM, which parameterizes the address dialog screen. Next, when you call FM ADDR_DIALOG (FM 384), it displays the changes you made to parameterize the address dialog box.

[↓] Import

FIELD_SELECTION

Field selection control template. Each character in this parameter string controls an address maintenance screen field group, depending on the character's position and the field group number. Example: the 22nd character controls the field to which "022" is assigned as the 4th modification screen group. Possible Control characters are – (minus, hide the field), (period, display field as optional field), + (plus, display field as required), * (asterisk, display only field), (Space, standard settings of field) and # (hash, field will be always visible on the standard maintenance screen). Type: ADDR1_FSEL-FISEL.

FIELD_SELECTION_FOR_NATION

Field for the selection control structure. Type: ADDR1_FSEL-FISEL.

USE_FS_FOR_NATION

Checkbox field. Type: SZAD_FIELD-FLAG.

KEYWORDS

Dynamic key word structure. Type: ADDR1_KEYW.

TITLEBAR

Address maintenance screen title bar. Type: SY-TITLE.

CHANGE_DEFAULT_COMM_TYPES

Activate/deactivate communication types. Type: ADDR_COMM.

FRAME_TEXT

Dynamic frame texts. Type: ADDR_FRAME.

DEFAULT_URI_TYPE

Default URI/URL/FTP type. Type: ADURI-URI_TYPE.

SCREEN_VARIANT

Screen (transaction) variant. Type: SHDFV-TCVARIANT.

SHOW_PUSHBOTTONS_AT_TOP

Display toolbar. Only valid for subscreen. Type: SZAD_FIELD-FLAG.

USE_PSEUDO_REQUIRED_FIELDS

Display required entry fields as pseudo-required entry fields. Type: XFELD.

IV_TIME_DEPENDENCE

Time-specific maintenance of communication data. Type: XFELD.

Tables

EXCLUDED_FUNCTIONS

GUI status functions to be excluded.

INCLUDED_FUNCTIONS

Additional functions to include.

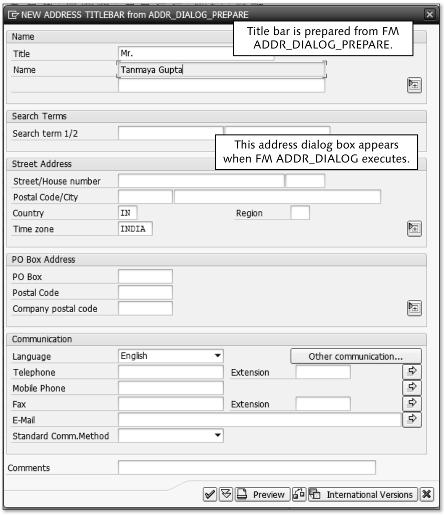
ERROR_TABLE

Table with errors, warnings, and information. Type: ADDR_ERROR.

[!] Exceptions

INTERNAL_ERROR

Serious internal error (MESSAGE A...).



NEW ADDRESS TITLEBAR from ADDR_DIALOG_PREPARE

Title bar is prepared from FM ADDR_DIALOG_PREPARE.

Title: Mr.

Name: Tanmaya Gupta

This address dialog box appears when FM ADDR_DIALOG executes.

Country: IN

Time zone: INDIA

Language: English

Address Dialog Box

386 ADDR_GET

This FM reads the address of an entity in the SAP system without the dialog box screen. If you want the details of the address and want to build your other program logic, you can call this FM to read the address details.

Example: To get the address for a plant on the selection screen, you can pass the parameter and call this FM.

[↓] Import

ADDRESS_SELECTION

Specification of an address. In the calling program, normally only the address number is recognized by a foreign key relationship (e.g., plant T001W-ADRNR, purchasing organization TVKO-ADRNR). The address number is to be passed to the parameter ADDRNUMBER.

Addresses created but not yet stored in the local memory can be identified and read with the parameter ADDRHANDLE. Only one of the two parameters ADDRNUMBER and ADDRHANDLE has to be filled.

This is a mandatory parameter. Type: ADDR1_SEL.

ADDRESS_GROUP

Address group. Entering an address group can trigger special handling: The system checks whether the data conversion from SADR to ADRC has already occurred for the addresses in the corresponding address group.

In any case, both return parameters ADDR1_VAL and SADR are filled.

If an address group whose addresses have already been converted is passed, ADRC is read, and the structure SADR is returned filled, in addition to ADDR1_VAL.

If no conversion has occurred, table SADR is read, and the SADR structure is returned; structure ADDR1_DATA from SADR is also filled. Type: ADRG-ADDR_GROUP.

READ_SADR_ONLY

Read in table SADR only. Type: SZAD_FIELD-FLAG.

READ_TEXTS

If this parameter is set to X, address texts are also read. Type: SZAD_FIELD-FLAG.

IV_CURRENT_COMM_DATA

This parameter determines the current status of communication data. Type: AD_COMCURR.

[↑] Export

ADDRESS_VALUE

Return data for an address. Type: ADDR1_VAL.

ADDRESS_ADDITIONAL_INFO

Additional communication data information. Type: AD1_FLAGS.

RETURNCODE

Return code. Space is success, I is information, W is warning, and E is error. Type: SZAD_FIELD-RETURNCODE.

ADDRESS_TEXT

Texts for key fields in the SY-LANGU language such as language key, country, and so on are returned if the parameter READ_TEXTS is set to X. Type: ADDR1_TEXT.

SADR

Return the old address structure. Type: SADR.

▦ Tables

ADDRESS_GROUPS

Address groups to which the address is assigned. Type: ADAGROUPS.

ERROR_TABLE

Table with errors, warnings, and information. Type: ADDR_ERROR.

VERSIONS

Table with international versions. Type: ADDR_VERS.

[!] Exceptions

PARAMETER_ERROR

Incorrect parameter values.

ADDRESS_NOT_EXIST

Address does not exist.

VERSION_NOT_EXIST

International version of the address does not exist.

INTERNAL_ERROR

Serious internal error (MESSAGE A...).

387 ADDR_GET_COMPLETE

This FM provides you with all of the details for the given address number or address handle, including all of the communication types and notes about the address or communication type. All of the valid communication types are defined in table TSAC(T).

Example: To get the complete address of the plant on your selection screen along with the communication details for that address, you can pass the parameter to and call this FM.

[↓] Import

ADDRNUMBER

Address number for which you want the complete information. This parameter is a key of the database table. Type: ADDR1_SEL-ADDRNUMBER.

ADDRHANDLE

Address handle for which you want the complete information. This is a temporary key for the address. Type: ADDR1_SEL-ADDRHANDLE.

ARCHIVE_HANDLE

Read address from archive. Type: SY-TABIX.

IV_CURRENT_COMM_DATA

This parameter determines the current status of communication data The default value is X. Type: AD_COMCURR.

[↑] Export

ADDR1_COMPLETE

The complete address object (including communication) is returned in this parameter. Type: SZADR_ADDR1_COMPLETE.

[!] Exceptions

PARAMETER_ERROR

Incorrect parameter values.

ADDRESS_NOT_EXIST

Address does not exist.

INTERNAL_ERROR

Serious internal error (MESSAGE A...).

WRONG_ACCESS_TO_ARCHIVE

Incorrect access to archived data.

388 ADDR_INSERT

This FM inserts an address without a dialog. This FM creates an address in the local memory of the function group SZA0. These FMs are also used in this process:

- ADDR_NUMBER_GET (FM 391)
- ADDR_MEMORY_SAVE (FM 390)
- ADDR_MEMORY_CLEAR (FM 389)

Example: To create a normal address, you can pass the relevant import parameter into and call this FM, which inserts the address in local memory.

[↓] Import

ADDRESS_DATA

Parameter with address data. Type: ADDR1_DATA.

ADDRESS_GROUP

Address group. Entering an address group can trigger special handling: The system checks whether the data conversion from SADR to ADRC has already occurred for the addresses in the corresponding address group. In any case, both return parameters ADDR1_VAL and SADR are filled.

If an address group whose addresses have already been converted is passed, ADRC is read, and structure SADR is returned filled, in addition to ADDR1_VAL.

If no conversion has occurred, table SADR is read, the SADR structure is returned, and structure ADDR1_DATA from SADR is also filled. Type: ADRG-ADDR_GROUP.

ADDRESS_HANDLE

Address handle for which you want the complete information. This is a temporary key for the address. Type: SZAD_FIELD-HANDLE.

DATE_FROM

Reference date for the address. Type: ADCP-DATE_FROM.

LANGUAGE

Language for the text box. Type: ADRCT-LANGU.

CHECK_EMPTY_ADDRESS

Address input must be checked. Type: SZAD_FIELD-FLAG.

CHECK_ADDRESS

If this parameter is set to X, the address contents are checked. The default value is X. Type: SZAD_FIELD-FLAG.

[↑] Export

ADDRESS_DATA

Parameter with address data. Type: ADDR1_DATA.

RETURNCODE

Return code of the address. if the value of the return code isn't equal to Space, at least one message is stored in the table ERROR_TABLE. Type: SZAD_FIELD-RETURNCODE.

▦ Tables

ERROR_TABLE

Table with errors, warnings, and information. Type: ADDR_ERROR.

[!] Exceptions

ADDRESS_EXISTS

Address already exists in the handle.

PARAMETER_ERROR

Incorrect parameter values.

INTERNAL_ERROR

Serious internal error (MESSAGE A...).

389 ADDR_MEMORY_CLEAR

This FM deletes the local memory contents. All address data changes made since the first call of any FM or the last local memory save are discarded. This FM should be called after extensive read operations or when returning to the application.

Example: You've modified the address from FM ADDR_INSERT (FM 388), which adds or modifies the address in the local memory. Now you're done with the modification of the address and want to clear the local memory so that it doesn't overflow when you want to modify another address. You can pass the parameter FORCE if you want unsaved data to be cleared as well or just execute the FM without passing any parameters. After this FM is executed, the local memory is cleared (refer to FM ADDR_INSERT [FM 388]).

[↓] Import

FORCE

If this parameter is X, unsaved data are also cleared. Type: SZAD_FIELD-FLAG.

[!] Exceptions

UNSAVED_DATA_EXIST

Missing address number for new address.

INTERNAL_ERROR

Some internal error occurred.

390 ADDR_MEMORY_SAVE

This FM saves all of the address data from the local memory to the database. All address data changes made since the first call of any FM in the function group are then permanent.

Example: You're creating an address from FM ADDR_INSERT (FM 388). To save the address to the respective database table, call this FM. Refer to FM ADDR_INSERT (FM 388).

[↓] Import

EXECUTE_IN_UPDATE_TASK

Database update in the dialog or in the update task. Type: SZAD_FIELD-FLAG.

[!] Exceptions

ADDRESS_NUMBER_MISSING

Missing address number for new address.

PERSON_NUMBER_MISSING

Missing person number for new person.

INTERNAL_ERROR

Serious internal error (MESSAGE A...).

DATABASE_ERROR

Error when writing to the database (dialog).

REFERENCE_MISSING

Reference is missing.

391 ADDR_NUMBER_GET

This FM assigns an address number; that is, converts a temporary address number (address handle) into a permanent address number, which is saved in the database key.

Example: You're creating a new address and need to assign the address number to the address information. This address number is a unique number that identifies the address. You can pass the relevant parameters in this

FM and execute it to get the address number and assign the numbers to the address. Refer to FM ADDR_INSERT (FM 388).

[↓] Import

ADDRESS_HANDLE

Address handle for which you want the complete information. This is a temporary key for the address. Type: SZAD_FIELD-HANDLE.

ADDRESS_REFERENCE

Address reference for the BAS where-used list. The consistency of the where-used list determines the reference integrity for deleting and archiving addresses and is also required for identifying the owner object of an address. Type: ADDR_REF.

PERSONAL_ADDRESS

Personal address indicator. The default value is Space. Type: ADRC-PERS_ADDR.

NUMBERRANGE_NUMBER

Number range from which the number is taken. Type: INRI-NRRANGENR.

E071K_WA

Prepopulated structure for the transport entry. Type: E071K.

GENERATE_TRANSPORT_ENTRIES

If this parameter is set to X, entries are generated for the transport request. Type: SZAD_FIELD-FLAG.

OWNER

This parameter indicates that the application object is the owner of the address. Type: SZAD_FIELD-FLAG.

TABLE_NAME

Referencing table for the transport entry. Type: TSADRV-DDIC_TABLE.

FIELD_NAME

Referencing table field for the transport entry. Type: TSADRV-DDIC_FIELD.

OBJKEY

Key of referencing table for the transport entry. Type: ADRCS2-OBJKEY.

[↑] Export

ADDRESS_NUMBER

Number taken from the number range. Type: ADRC-ADDRNUMBER.

RETURNCODE_NUMBERRANGE

Return code when taking the number. Type: INRI-RETURNCODE.

▦ Tables

E071K_TAB

All transport entries that belong to the address. Type: E071K.

[!] Exceptions

ADDRESS_HANDLE_NOT_EXIST

Transferred address handle does not exist.

INTERNAL_ERROR

Serious internal error (MESSAGE A...).

PARAMETER_ERROR

Incorrect parameter values.

392 ADDRESS_INTO_PRINTFORM

This FM formats an address for printing in the correct postal form with the required number of lines from the input. This FM also returns a short address; that is, the name and city one-line short address for the data carrier. Many country-specific formats are supported that correspond to the postal regulations of recipient countries.

Example: If your application isn't using the Business Address Services (BAS) for addresses, but you still want to have country-specific addresses displayed in your form, you can do pass the relevant parameters into and call this FM. When you call this FM, you'll receive the country-specific address that can be used for printing the address.

[↓] Import

ADRSWA_IN

Address data/control parameter. This parameter is only used in address type Space. Type: ADRS.

ADDRESS_1

Address data transfer structure for type 1. Type: ADRS1.

ADDRESS_2

Address data transfer structure for type 2. Type: ADRS2.

ADDRESS_3

Address data transfer structure for type 3. Type: ADRS3.

ADDRESS_TYPE

There are three address types:

- For address type 1: The normal address (company or organization) structure used in most SAP applications.
- For address type 2: Address of a person.
- For address type 3: Work address, which is usually the address of a contact person in a company.

The default address type Space is treated as type 1 and is required for the upward compatibility of the FM.

The interface parameters ADRSWA_IN and ADRSWA_OUT are only used in address type Space. Type: SZAD_FIELD-ADDR_TYPE.

ADDRESS_NUMBER

Address number (data is read directly). The address fields (Name, Street, City, etc.) for formatting can either be passed in the appropriate structure (ADDRESS1/2/3 or ADRSWA_IN) or be read from the BAS database tables, if this address is there, using the address key. To read the database tables, the keys are:

- For address type 1: ADDRESS_NUMBER
- For address type 2: ADDRESS_NUMBER and PERSON_NUMBER
- For address type 3: ADDRESS_NUMBER and PERSON_NUMBER.

Type: ADRC-ADDRNUMBER.

ADDRESS_HANDLE

Address handle for which you want the complete information. This is a temporary key for the address. Type: SZAD_FIELD-HANDLE.

PERSON_NUMBER

Person number (data are read directly). Type: ADRP-PERSNUMBER.

PERSON_HANDLE

Person handle for which you want the complete information. Type: SZAD_FIELD-HANDLE.

SENDER_COUNTRY

Sender country (for type 1, 2, and 3). Type: SZAD_FIELD-SEND_CNTRY.

RECEIVER_LANGUAGE

Recipient language (for type 1, 2, and 3). Type: T002-SPRAS.

NUMBER_OF_LINES

Max. line length for the formatting (for type 1, 2, and 3). The default value is 10. Type: ADRS-ANZZL.

STREET_HAS_PRIORITY

If this parameter is set to X, Street has priority over PO Box. Type: SZAD_FIELD-STREETPRIO.

LINE_PRIORITY

Line suppression sequence. Type: SZAD_FIELD-PRIORITY.

COUNTRY_NAME_IN_RECEIVER_LANGU

If this parameter is set to X, the country outputs in the recipient language. Type: SZAD_FIELD-FLAG.

LANGUAGE_FOR_COUNTRY_NAME

This parameter specifies the country output language for a foreign address. Otherwise, the sender country language is used. Type: T002-SPRAS.

NO_UPPER_CASE_FOR_CITY

International post guidelines specify that the city and country in foreign addresses are output in uppercase.

If this parameter is set to X, there is no conversion to uppercase. If Space, country and city are converted to uppercase. Type: SZAD_FIELD-FLAG.

IV_NATION

International version for formatting. Type: ADRC-NATION.

IV_NATION_SPACE

Flag: Use international version space in all cases. Type: T_BOOLE.

IV_PERSON_ABOVE_ORGANIZATION

Print person above organization. Type: T_BOOLE.

IS_BUPA_TIME_DEPENDENCY

Transfer structure for time dependency. Type: ADBUPA_TD.

IV_LANGU_CREA

Language key. Type: T002-SPRAS.

IV_DISPLAY_COUNTRY_IN_SHRTFORM

Display country key in short form. Type: XFELD.

[↑] Export

ADRSWA_OUT

This parameter is only used in address type Space.

The formatted address is returned in the 10 fields LINE0 to LINE9 in the return structure ADRSWA_OUT.

A one-line short address comprising the name and city is returned in the field LINEK. The format is NAME1/City line (from the formatted address).

A one-line short address for "Payment without documents by data carrier exchange (DCE)" is returned in the field LINED. This usually comprises the postal code, city, and country, and is formatted specific to the country. Type: ADRS.

ADDRESS_PRINTFORM

Formatted address (type 1, 2, and 3). Type: ADRS_PRINT.

ADDRESS_SHORT_FORM

Short form of the formatted address (type 1, 2, and 3). Type: SZAD_FIELD-ADDR_SHORT.

ADDRESS_SHORT_FORM_S

Short form, including street (type 1, 2, and 3). Type: SZAD_FIELD-ADDR_SHORT.

ADDRESS_DATA_CARRIER

Data carrier address (type 1, 2, and 3). Type: SZAD_FIELD-ADDR_DC.

ADDRESS_DATA_CARRIER_0

Second line for the data medium (type 1, 2, and 3). This line is part of the two-line data medium address. The line is the same as the address line, which contains street or PO box. Type: SZAD_FIELD-ADDR_DC.

NUMBER_OF_USED_LINES

Number of lines occupied (type 1, 2, and 3). Type: ADRS-ANZZL.

NAME_IS_EMPTY

Flag: Name fields empty, no formatting. Type: SZAD_FIELD-FLAG.

ADDRESS_NOT_FOUND

This parameter has the value X if the address isn't found in the BAS database table when the address number is entered. Type: SZAD_FIELD-FLAG.

ADDRESS_PRINTFORM_TABLE

Formatted address in table form. The formatted address is returned in an internal table. LINE_TYPE specifies the line type (name, city, street, etc.) per line. Type: SZADR_PRINTFORM_TABLE.

ADDRESS_SHORT_FORM_WO_NAME

Short form street and city (type 1, 2, and 3). Type: SZAD_FIELD-ADDR_SHORT.

EV_NATION

Actual version used if IV_NATION was transmitted. Type: ADRC-NATION.

393 ADDRESS_SHOW_PRINTFORM

This FM displays the formatted address in a popup.

Example: To display the print preview of an address according to the postal guidelines of a country, call this FM and fill in the parameters, and the address appears in a popup.

[↓] Import

ADDRESS_1

Address data transfer structure for type 1. Type: ADRS1.

ADDRESS_2

Address data transfer structure for type 2. Type: ADRS2.

ADDRESS_3

Address data transfer structure for type 3. Type: ADRS3.

ADDRESS_TYPE

There are three address types:

- For address type 1: Normal address (company or organization) structure used in most SAP applications.
- For address type 2: Address of a person.
- For address type 3: Work address, which is usually the address of a contact person in a company.

The default address type Space is treated as type 1 and is required for the upward compatibility of the FM.

The interface parameters ADRSWA_IN and ADRSWA_OUT are only used in address type Space. Type: SZAD_FIELD-ADDR_TYPE.

ADDRESS_NUMBER

Address number (data is read directly). The address fields (Name, Street, City, etc.) for formatting can either be passed in the appropriate structure (ADDRESS1/2/3 or ADRSWA_IN) or be read from the BAS database tables, if this address is there, using the address key. To read the database tables, the keys are:

- For address type 1: ADDRESS_NUMBER
- For address type 2: ADDRESS_NUMBER and PERSON_NUMBER

- For address type 3: ADDRESS_NUMBER and PERSON_NUMBER.

Type: ADRC-ADDRNUMBER.

ADDRESS_HANDLE

Address handle for which you want the complete information. This is a temporary key for the address. Type: SZAD_FIELD-HANDLE.

PERSON_NUMBER

Person number (data are read directly). Type: ADRP-PERSNUMBER.

PERSON_HANDLE

Person handle for which you want the complete information. Type: SZAD_FIELD-HANDLE.

SENDER_COUNTRY

Sender country (for type 1, 2, and 3). Type: SZAD_FIELD-SEND_CNTRY.

RECEIVER_LANGUAGE

Recipient language (for type 1, 2, and 3). Type: T002-SPRAS.

NUMBER_OF_LINES

Max. line length for the formatting (type 1, 2, and 3). The default value is 10. Type: ADRS-ANZZL.

STREET_HAS_PRIORITY

If this parameter is set to X, street has priority over PO box. Type: SZAD_FIELD-STREETPRIO.

LINE_PRIORITY

Line suppression sequence. Type: SZAD_FIELD-PRIORITY.

COUNTRY_NAME_IN_RECEIVER_LANGU

If this parameter is set to X, country outputs in the recipient language. Type: SZAD_FIELD-FLAG.

LANGUAGE_FOR_COUNTRY_NAME

This parameter specifies the country output language for a foreign address. Otherwise, the sender country language is used. Type: T002-SPRAS.

NO_UPPER_CASE_FOR_CITY

International post guidelines specify that the city and country in foreign addresses are output in uppercase.

If this parameter is set to X, there is no conversion to uppercase. If Space, country and city are converted to uppercase. Type: SZAD_FIELD-FLAG.

IS_BUPA_TIME_DEPENDENCY

Transfer structure for time dependency. Type: ADBUPA_TD.

[↑] Export

ADDRESS_PRINTFORM

Formatted address (type 1, 2, and 3). Type: ADRS_PRINT.

ADDRESS_SHORT_FORM

Short form of the formatted address (type 1, 2, and 3). Type: SZAD_FIELD-ADDR_SHORT.

ADDRESS_DATA_CARRIER

Data carrier address (type 1, 2, and 3). Type: SZAD_FIELD-ADDR_DC.

NUMBER_OF_USED_LINES

Number of lines occupied (type 1, 2, and 3). Type: ADRS-ANZZL.

NAME_IS_EMPTY

Flag: Name fields empty, no formatting. Type: SZAD_FIELD-FLAG.

ADDRESS_NOT_FOUND

This parameter has the value X if the address isn't found in the BAS database table when the address number is entered. Type: SZAD_FIELD-FLAG.

EV_EXIT_CAUSED_BY_CALLER

Delete list in CALLBACK_USER_COMMAND. Type: CHAR1.

EV_EXIT_CAUSED_BY_USER

How the user left the list. Type: SLIS_EXIT_BY_USER.

394 AUTHORITY_CHECK_DATASET

This FM allows you to check authorizations to access files with the key words OPEN DATASET, READ DATASET, TRANSFER, and DELETE DATASET. The authorization check is performed using the authorization object S_DATASET.

Example: You want to access the file *ABC*. Before opening the file, first you need to check if you're authorized to access the file. You can pass the parameters into and call FM AUTHORITY_CHECK_DATASET. You'll get an exception if you're not authorized to access the file.

[↓] Import

PROGRAM

ABAP program from which authorization needs to be checked. If no program name is specified, the system assumes the current program. Type: AUTHB-PROGRAM.

ACTIVITY

Access type. Possible values are:

- READ: Read file.
- WRITE: Change file.
- READ_WITH_FILTER: Read file with filter function.
- WRITE_WITH_FILTER: Change file with filter function.
- DELETE: Delete file.

Type: C.

FILENAME

File name for authorization. Type: AUTHB-FILENAME.

[!] Exceptions

NO_AUTHORITY

You are not authorized for this access.

ACTIVITY_UNKNOWN

Access type unknown.

395 AUTHORITY_CHECK_TCODE

This FM checks whether you're authorized to execute a transaction code. The authorization check is performed using the authorization object S_TCODE. S_TCODE is the authorization object linked with the transaction code authorization.

Example: You want your transaction to be executed only by authorized users. You need to pass the parameters and call this FM in the `INITILIZATION` event so that the transaction can be loaded only for the authorized users. You can also use the ABAP command `AUTORITY-CHECK OBJECT` for the same purpose.

[↓] Import

TCODE

Transaction code. Type: TSTC-TCODE.

[!] Exceptions

OK

Authorized for the transaction.

NOT_OK

Not authorized for the transaction.

396 SUSR_USER_AUTH_FOR_OBJ_GET

This FM displays authorizations for a specified object belonging to a user. User authorizations are displayed in an internal table. It's also possible to restrict the authorizations to specified authorization objects.

Example: You want to know whether you have authorization to start Transaction SU01. The user requires authorization for object S_TCODE with the value SU01 in field TCD to be able to do this. You can pass the parameters into and call FM SUSR_USER_AUTH_FOR_OBJ_GET. The value is returned in the table parameter VALUES if you have authorization.

[↓] Import

NEW_BUFFERING

This parameter is used as an indicator for various authorization programs. Type: XUFLAG.

MANDANT

SAP system and client number. Type: SY-MANDT.

USER_NAME

User name in user master record. This is a mandatory parameter. Type: USR02-BNAME.

SEL_OBJECT

Authorization object for restriction. If this parameter is passed, only the permissions to the object SEL_OBJECT are displayed. Type: UST12-OBJCT.

⊞ Tables

VALUES

Authorization values are returned in this table. Type: USVALUES.

[!] Exceptions

USER_NAME_NOT_EXIST

User name does not exist.

NOT_AUTHORIZED

No authorization.

INTERNAL_ERROR

Internal error.

Index

A

ABAP
- *download keyword document information*, 46
- *download spool list details*, 531
- *spool list*, 535
- *statement MESSAGE*, 561

ABAP Data Dictionary
- *create field catalog*, 276
- *create/activate table*, 92
- *field catalog for ALV*, 269
- *FIELDS*, 570
- *key word*, 569
- *read data element*, 86
- *read domain*, 85
- *read table*, 92
- *read type*, 87, 89
- *read view*, 95
- *save inbound IDoc in DB tables*, 480
- *specifying fields*, 569
- *transaction tables*, 297
- *validate/check table field values*, 574
- *write table information*, 94

ABAP Data Dictionary type
- *get runtime information*, 89

ABAP field
- *read from container*, 937
- *write in container*, 937

ABAP memory
- *retrieve list*, 267

ABAP program
- *get list of global data*, 913
- *return list of all global data*, 913

ABAP source code
- *rearrange*, 54

ABAP Workbench, 306
- *TADIR*, 613

ABAP_DOCU_DOWNLOAD, 46

ABAP4_CALL_TRANSACTION, 45

Account balance
- *return from key date*, 683

Accounting
- *billing document*, 660

Accounting (Cont.)
- *post/reverse manual cost allocation*, 675
- *update business transaction*, 669
- *validate business transactions*, 668

Accounting period
- *determine first/last day*, 218

ActiveX
- *GUI supported controls*, 433

Activity allocation document
- *post*, 659

Actual data
- *post to SAP*, 718

ADD_TIME_TO_DATE, 147, 232

ADDR_DIALOG, 487

ADDR_DIALOG_PREPARE, 488

ADDR_GET, 489

ADDR_GET_COMPLETE, 491

ADDR_INSERT, 492

ADDR_MEMORY_CLEAR, 493

ADDR_MEMORY_SAVE, 494

ADDR_NUMBER_GET, 494

Address
- *change*, 688
- *convert address handle into a permanent address number*, 494
- *create in local memory SZA0*, 492
- *data save from local memory to DB*, 494
- *delete for business partner*, 693
- *dialog*, 488
- *discarded data changes*, 493
- *display dialog box to maintain address*, 487
- *display formatted address in a popup*, 499
- *format for printing*, 496
- *get complete details for a given address number/handle*, 491
- *insert address without a dialog*, 492
- *maintenance*, 488
- *number*, 494
- *read address without dialog*, 489
- *read for business partner*, 691
- *return short address for data carrier*, 496
- *set parameters for address maintenance*, 488

Address dialog screen, reduce parameters, 488
ADDRESS_INTO_PRINTFORM, 496
ADDRESS_SHOW_PRINTFORM, 499
ADK, 389
  give data statistics to, 392
Administration information, 459
Administrator function
  log off user, 71
Adobe Form
  FM name, 347
ALE, 38, 463
  determine BAPI logical receiver system, 465
  determine receiver for synchronous BAPI, 468
  distribute users, 854
ALE distribution model, 465
  filter object types of BAPIs, 466
ALE_ASYNC_BAPI_GET_RECEIVER, 465
ALE_BAPI_GET_FILTEROBJECTS, 466
ALE_BAPI_GET_UNIQUE_RECEIVER, 466
ALE_IDOCS_CREATE, 467
ALE_SYNC_BAPI_GET_RECEIVER, 468
Alpha conversion, 154
ALSM_EXCEL_TO_INTERNAL_TABLE, 423
ALV
  block list, display, 274
  grid format, 277
  initialize block, 274
  output, display, 269
  read default variant, 270, 291
  read display variant, 294
  report, generate, 437
  top-of-page, 275
  variant, 4F help, 293
ALV list
  blocked, 272
  change, 283
  change information, 289
  create report, 286
  get hierarchy, 283
  get info for current list, 286
Amount
  convert to words, 414
  set places around decimal, 414
APPL_LOG_DELETE, 621
APPL_LOG_DISPLAY, 622
APPL_LOG_DISPLAY_INTERN, 623
APPL_LOG_INIT, 624
APPL_LOG_INIT_MESSAGES, 624
APPL_LOG_READ_DB, 625
APPL_LOG_READ_INTERN, 626
APPL_LOG_SET_OBJECT, 628
APPL_LOG_WRITE_DB, 629
APPL_LOG_WRITE_HEADER, 629
APPL_LOG_WRITE_LOG_PARAMETERS, 630
APPL_LOG_WRITE_MESSAGE_PARAMS, 631
APPL_LOG_WRITE_MESSAGES, 632
APPL_LOG_WRITE_SINGLE_MESSAGE, 631, 632, 633
Application
  give authorization to parent program, 644
Application log, 40, 619
  add message to application log, 651
  analyze local memory, 623
  assign authorization to perform critical functions, 644
  check if specified subobject exists, 628
  configure at start of transaction, 646
  delete all logs from main memory, 647
  delete all protocols, 621
  delete existing entries for object/subobject in local memory, 624
  delete log header and messages from memory, 655
  delete message, 653
  display, 637
  display in standard transaction format, 643
  display messages in popup, 641
  display selected, 622
  initialize local memory, 624
  insert free text with log handle, 652
  open header data, 649
  print, 639
  read from database/load to memory, 625
  read messages and text, 654
  read object/subobject data from local memory, 626
  refresh entire memory, 647
  reset authorization key, 645
  return existing configuration, 645

Application log (Cont.)
  return log handle from I_S_LOG_HEADER, 649
  return log handler, 649
  return number of read logs, 627
  return profile from transaction SLG1, 643
  search message in memory, 648
  write data from local memory, 629
  write header data to local memory, 629
  write message with parameters in local memory, 631
  write multiple messages without parameters to local memory, 632
  write new log for object/subobject with same header data, 624
  write single message without parameters, 633
Application server
  F4 help for files, 552
  give list of registered, 76
Archive
  close open files, 383
  open to delete data, 393
  open to read, 397
  return file for index-based access of data, 406
Archive Development Kit -> see ADK, 389
Archive file
  create new, 398
  read, 395
Archive handle
  open, 384
  read next data object, 386
Archive information system
  read data, 406
Archive parameters
  define/modify/display, 518
ARCHIVE_CLOSE_FILE, 383
ARCHIVE_GET_CUSTOMIZING_DATA, 383
ARCHIVE_GET_INFORMATION, 384
ARCHIVE_GET_NEXT_OBJECT, 386
ARCHIVE_GET_NEXT_RECORD, 387, 402
ARCHIVE_GET_STATISTICS, 389
ARCHIVE_GET_TABLE, 390, 406
ARCHIVE_GIVE_STATISTICS, 392
ARCHIVE_NEW_OBJECT, 393
ARCHIVE_OPEN_FOR_DELETE, 393
ARCHIVE_OPEN_FOR_MOVE, 395
ARCHIVE_OPEN_FOR_READ, 397, 883, 885, 886
ARCHIVE_OPEN_FOR_WRITE, 398, 404
ARCHIVE_PUT_RECORD, 400
ARCHIVE_PUT_TABLE, 401
ARCHIVE_READ_OBJECT, 402, 406
ARCHIVE_READ_STATISTICS, 405
ARCHIVE_SAVE_OBJECT, 404
ARCHIVE_WRITE_STATISTICS, 405
Archiving
  class, 386
  transfer statistics, 389
Archiving object
  get customizing data, 384
  get handle information, 384
  give settings to archiving program, 383
ARFC_GET _TID, 47
Article, change reservation, 824
AS_API_READ, 406
ASCII data
  transfer to start Excel, 440
ASCII file
  tranfer to Excel format, 439
Asset Accounting
  post asset retirement in financial accounting, 685
Asset master record
  change, 732
  create, 735
Asset retirement
  post in asset accounting, 685
ATP logic, 751
Attachment
  create for existing document, 443
  return all attributes, 445
Authority check
  check file access authorization, 501
  check transaction execution authorization, 502
  display specified object authorization, 502
AUTHORITY_CHECK_DATASET, 501
AUTHORITY_CHECK_RFC, 47
AUTHORITY_CHECK_TCODE, 502
Authorization
  assign for application log critical function, 644

Authorization (Cont.)
check, 47, 128
check to open file with keyword, 501
check to table, 128
check to view, 128
check user for executing commands, 65
check user for text module, 327
check user to execute external command, 67
object belonging to user, 502
reset key, 645
Authorization object
S_DATASET, 501
S_TCODE, 502

B

Background event
trigger, 48
Background job
close, 869
close requests, 521
create, 525
delete aborted, 507
find request, 507
pass to background processing system, 521
processing, complete, 868
read runtime data, 517
Background processing, 505
creates job from external system, 869
get status of job, 870
interface, 514
Background processing system
start background job, 48
BAL_DB_DELETE, 634
BAL_DB_LOAD, 635, 637
BAL_DB_SAVE, 636
BAL_DB_SEARCH, 634, 635, 637
BAL_DSP_LOG_DISPLAY, 637, 641, 643, 654
BAL_DSP_LOG_PRINT, 639
BAL_DSP_PROFILE_*, 639
BAL_DSP_PROFILE_DETLEVEL_GET, 640
BAL_DSP_PROFILE_NO_TREE_GET, 641
BAL_DSP_PROFILE_POPUP_GET, 641
BAL_DSP_PROFILE_SINGLE_LOG_GET, 642
BAL_DSP_PROFILE_STANDARD_GET, 643
BAL_GLB_AUTHORIZATION_GET, 644, 646
BAL_GLB_AUTHORIZATION_RESET, 644, 645
BAL_GLB_CONFIG_GET, 645
BAL_GLB_CONFIG_SET, 644, 646
BAL_GLB_MEMORY_REFRESH, 644
BAL_GLB_SEARCH_LOG, 636, 647, 648
BAL_GLB_SEARCH_MSG, 648
BAL_LOG_CREATE, 649
BAL_LOG_HDR_READ, 650
BAL_LOG_MSG_ADD, 651
BAL_LOG_MSG_ADD_FREE_TEXT, 652
BAL_LOG_MSG_DELETE, 653
BAL_LOG_MSG_DELETE_ALL, 654, 655
BAL_LOG_MSG_READ, 648, 654
BAL_LOG_REFRESH, 654, 655
BAPI
asynchronous, 465, 467
characteristics class hierarchy, 704
determine ALE distribution model, 466
execute ROLLBACK WORK, 850
F4 help, 741
parameter, 741
read and get details of planned order, 785
synchronous, 466, 468
BAPI_ACC_ACTIVITY_ALLOC_POST, 659
BAPI_ACC_BILLING_POST, 660
BAPI_ACC_DOCUMENT_CHECK, 662, 665, 666, 667, 669, 670, 674, 675
BAPI_ACC_DOCUMENT_POST, 663, 665
BAPI_ACC_DOCUMENT_REV_POST, 667, 674
BAPI_ACC_GL_POSTING_CHECK, 668
BAPI_ACC_GL_POSTING_POST, 668, 669
BAPI_ACC_GL_POSTING_REV_POST, 670
BAPI_ACC_INVOICE_RECEIPT_CHECK, 671
BAPI_ACC_INVOICE_RECEIPT_POST, 671, 673
BAPI_ACC_MANUAL_ALLOC_POST, 675
BAPI_ALM_CONF_CREATE, 676
BAPI_ALM_NOTIF_CREATE, 677
BAPI_ALM_NOTIF_SAVE, 677, 679
BAPI_ALM_ORDER_GET_DETAIL, 679
BAPI_ALM_ORDER_MAINTAIN, 681
BAPI_AR_ACC_GETKEYDATEBALANCE, 683
BAPI_AR_ACC_GETOPENITEMS, 684
BAPI_AR_ACC_GETPERIODBALANCES, 685

BAPI_ASSET_RETIREMENT_POST, 685
BAPI_BILLINGDOC_CANCEL1, 686
BAPI_BILLINGDOC_CREATEMULTIPLE, 687
BAPI_BUPA_ADDRESS_CHANGE, 688, 693
BAPI_BUPA_ADDRESS_GETDETAIL, 691
BAPI_BUPA_ADDRESS_REMOVE, 693
BAPI_BUPA_CENTRAL_CHANGE, 693
BAPI_BUPA_CENTRAL_GETDETAIL, 696
BAPI_BUPA_CREATE_FROM_DATA, 698
BAPI_BUS2001_SET_STATUS, 812, 871
BAPI_CHARACT_CREATE, 700, 704
BAPI_CHARACT_GETDETAIL, 701
BAPI_CLASS_CHANGE, 702, 705
BAPI_CLASS_CREATE, 700, 704, 706
BAPI_CLASS_GET_CHARACTERISTICS, 703, 705
BAPI_CLASS_GET_DETAILS, 703
BAPI_CLASS_GETDETAIL, 706
BAPI_CLASS_GETDETAILS, 707
BAPI_CLASS_GETSTATUS, 707
BAPI_COMPANYCODE_GET_PERIOD, 708
BAPI_COMPANYCODE_GETLIST, 709
BAPI_CONTRACT_CHANGE, 709, 715
BAPI_CONTRACT_CREATE, 712
BAPI_CONTRACT_GETDETAIL, 709, 715
BAPI_COPAACTUALS_POSTCOSTDATA, 718
BAPI_COPAPLANNING_GETDATA, 719
BAPI_COPAPLANNING_POSTDATA, 719, 720
BAPI_CURRENCY_CONV_TO_EXTERNAL, 182, 721
BAPI_CURRENCY_CONV_TO_INTERNAL, 721
BAPI_CUSTOMER_CHANGEFROMDATA1, 722
BAPI_CUSTOMER_CREATE, 723
BAPI_CUSTOMER_GETLIST, 724
BAPI_CUSTOMERINQUIRY_CHANGE, 724
BAPI_CUSTOMERQUOTATION_CHANGE, 727
BAPI_ENTRYSHEET_CREATE, 729
BAPI_EQUI_CHANGE, 730
BAPI_EQUI_CREATE, 731
BAPI_FIXEDASSET_CHANGE, 732
BAPI_FIXEDASSET_CREATE1, 735
BAPI_GOODSMVMT_GETITEMS, 738
BAPI_GOODSMVT_CANCEL, 738
BAPI_GOODSMVT_CREATE, 739
BAPI_GOODSMVT_GETITEMS, 740
BAPI_HELPVALUES_GET, 741
BAPI_HU_CREATE, 743
BAPI_HU_PACK, 744
BAPI_HU_UNPACK, 744
BAPI_IBAN_CHANGE, 745
BAPI_INB_DELIVERY_CONFIRM_DEC, 746
BAPI_INSPECTIONPLAN_CREATE, 748
BAPI_INTERNALORDER_CREATE, 750
BAPI_MATERIAL_AVAILABILITY, 751
BAPI_MATERIAL_BOM_GROUP_CREATE, 753
BAPI_MATERIAL_DELETE, 754
BAPI_MATERIAL_EDIT, 754
BAPI_MATERIAL_GET_ALL, 755
BAPI_MATERIAL_GET_DETAIL, 755, 757
BAPI_MATERIAL_GETLIST, 758
BAPI_MATERIAL_SAVEDATA, 759
BAPI_MATERIAL_SAVEREPLICA, 762
BAPI_MATERIAL_STOCK_REQ_LIST, 765
BAPI_NETWORK_COMP_ADD, 768
BAPI_OBJCL_CHANGE, 769, 772, 773
BAPI_OBJCL_CREATE, 770
BAPI_OBJCL_GETCLASSES, 771
BAPI_OBJCL_GETDETAIL, 772
BAPI_OBJCL_GETDETAIL_KEY, 773
BAPI_OBJCL_GETSTATUS, 774
BAPI_OUTB_DELIVERY_CONFIRM_DEC, 775
BAPI_OUTB_DELIVERY_CREATE_SLS, 778
BAPI_OUTB_DELIVERY_CREATE_STO, 779
BAPI_PLANNEDORDER_CHANGE, 781
BAPI_PLANNEDORDER_CREATE, 782
BAPI_PLANNEDORDER_DELETE, 783
BAPI_PLANNEDORDER_EXIST_CHECK, 784
BAPI_PLANNEDORDER_GET_DET_LIST, 786
BAPI_PLANNEDORDER_GET_DETAIL, 785
BAPI_PO_CHANGE, 786, 795
BAPI_PO_CREATE1, 791
BAPI_PO_GETDETAIL1, 795
BAPI_PO_GETITEMS, 798
BAPI_PO_RELEASE, 799
BAPI_PO_RESET_RELEASE, 801
BAPI_PR_CHANGE, 802

BAPI_PR_CREATE, 804
BAPI_PRODORD_CHANGE, 807
BAPI_PRODORD_CREATE, 808
BAPI_PRODORDCONF_CREATE_HDR, 809
BAPI_PRODORDCONF_CREATE_TT, 810
BAPI_PS_INITIALIZATION, 812, 871
BAPI_PS_PRECOMMIT, 812, 871
BAPI_QUOTATION_CREATEFROMDATA2, 813
BAPI_REQUIREMENT_CREATE, 815
BAPI_REQUIREMENTS_CHANGE, 816
BAPI_REQUIREMENTS_CREATE, 818
BAPI_REQUISITION_CHANGE, 819
BAPI_REQUISITION_CREATE, 820
BAPI_REQUISITION_DELETE, 822
BAPI_REQUISITION_RELEASE, 822
BAPI_RESERVATION_CHANGE, 824, 826
BAPI_RESERVATION_CREATE1, 825
BAPI_RESERVATION_GETDETAIL1, 826
BAPI_RESERVATION_GETDETAILS1, 824
BAPI_ROUTING_CREATE, 827
BAPI_SAG_CHANGE, 829, 836
BAPI_SAG_CREATE, 832
BAPI_SAG_GETDETAIL, 829, 836
BAPI_SALESORDER_CHANGE, 839, 849, 850
BAPI_SALESORDER_CREATEFROMDAT2, 841
BAPI_SALESORDER_GETLIST, 844
BAPI_SALESORDER_SIMULATE, 845
BAPI_SCHEDULE_MAINTAIN, 847
BAPI_SOURCEDETERMIN_GETSOS, 848
BAPI_TRANSACTION_COMMIT, 676, 685, 703, 704, 812, 849
BAPI_TRANSACTION_ROLLBACK, 850
BAPI_USER_ACTGROUPS_ASSIGN, 850
BAPI_USER_ACTGROUPS_DELETE, 851
BAPI_USER_CHANGE, 852
BAPI_USER_CLONE, 854
BAPI_USER_CREATE1, 857
BAPI_USER_DELETE, 859
BAPI_USER_EXISTENCE_CHECK, 860
BAPI_USER_GET_DETAIL, 860, 863
BAPI_USER_GETLIST, 862
BAPI_USER_LOCK, 863, 864
BAPI_USER_PROFILES_ASSIGN, 864
BAPI_USER_PROFILES_DELETE, 865
BAPI_USER_UNLOCK, 863, 864
BAPI_USR01DOHR_GETEMPLOYEE, 865
BAPI_VENDOR_CREATE, 866
BAPI_VENDOR_DELETE, 867
BAPI_VENDOR_GETDETAIL, 867
BAPI_WHSE_STOCK_GET_DETAIL, 868
BAPI_XBP_JOB_CLOSE, 868, 869
BAPI_XBP_JOB_OPEN, 869
BAPI_XBP_JOB_STATUS_GET, 870
BAPI-TRANSSACTION_COMMIT, 677
Base unit exponents
  determine dimension key, 196
Basis-related operations, 43
Batch input data
  create, 914
  insert to BDC session, 249
BDC
  create new session, 250
  insert batch input data, 249
BDC_CLOSE_GROUP, 249, 250
BDC_INSERT, 249, 250
BDC_OPEN_GROUP, 249, 250
Billing document
  cancel, 686
  create, 687
  post in accounting, 660
Binary format
  convert to XString, 226
Block list data, 272
BOM
  display simple material, 897
  explode for equipment item, 889
  explode for material, 892
  find where-used list, 896
  find where-used list of materials, 900
  return list of valid engineering change numbers, 900
  select material, plant, or BOM usage, 899
BOM explosion
  planned order, 782
BP_EVENT_RAISE, 48
BP_FIND_JOBS_WITH_PROGRAM, 507
BP_JOB_DELETE, 507
BP_JOB_READ, 509
BP_JOB_SELECT, 510
BP_JOBLOG_READ, 512

BP_JOBLOG_SHOW, 513
BP_START_DATE_EDITOR, 514
BR* tool
- *environment variable value*, 72

Business object
- *call/start method*, 941
- *instantiate from program*, 940
- *release method*, 942
- *trigger event*, 938

Business partner
- *change address*, 688
- *change central/general data*, 693
- *create*, 698
- *delete address/data*, 693
- *read address/communication data*, 691
- *read central attributes*, 696

Business transaction
- *update*, 669
- *validate*, 668

C

C14B_ADD_TIME, 148
CALCULATE_EXCHANGE_RATE, 148
Calendar date
- *determine from fiscal year*, 213
- *find day/month/year between*, 202

Calendar day
- *add/subtract*, 209
- *return attributes*, 191

Calendar quarter, 161, 162
CALL_BROWSER, 875
CAT_CHECK_RFC_DESTINATION, 49
Central User Administration (CUA), 854
CF_UT_UNIT_CONVERSION, 150, 214
Change
- *commit to database*, 50
- *save made in maintenance/service notifications*, 679

Change and Transport System, 598
Change document, 875
- *close creation*, 875
- *complete the processing*, 875
- *create items*, 879
- *create single item*, 888
- *delete for clients*, 877

Change document (Cont.)
- *display*, 878
- *edit table*, 881
- *interface*, 469
- *open creation*, 880
- *process old/new items according to category*, 882
- *read header from DB/archive*, 884
- *read headers*, 882
- *read items*, 882
- *read/process new values by category*, 886
- *write*, 469
- *write header*, 875

Change pointer
- *list*, 470
- *read*, 469
- *read processed*, 469
- *read unprocessed*, 469
- *status table*, 470
- *write status record*, 470

CHANGE_POINTERS_READ, 469
CHANGE_POINTERS_STATUS_WRITE, 470
CHANGEDOCUMENT_CLOSE, 875, 880
CHANGEDOCUMENT_DELETE, 877, 880
CHANGEDOCUMENT_DISPLAY, 878, 880
CHANGEDOCUMENT_MULTIPLE_CASE, 879, 881
CHANGEDOCUMENT_OPEN, 876, 880, 888
CHANGEDOCUMENT_PREPARE_TABLES, 879, 881
CHANGEDOCUMENT_READ, 882
CHANGEDOCUMENT_READ_HEADERS, 884
CHANGEDOCUMENT_READ_POSITIONS, 886
CHANGEDOCUMENT_SINGLE_CASE, 876, 880, 888
CHAR_FLTP_CONVERSION, 151
Character format, 204
Character name
- *convert style/form*, 343

Character string
- *put multiple in one*, 415

Character type
- *convert to float type*, 151

Characteristics
change class/data, 702
create new, 700
group in class, 704
read values, 705
return details, 701
Check total, 518
CHECK_TEXT_AUTHORITY, 327
CHECK_TEXT_ID, 327
CHECK_TEXT_LANGUAGE, 328
CHECK_TEXT_NAME, 329
CHECK_TEXT_OBJECT, 329
Class
CL_GUI_FRONTEND_SERVICES, 434
get details, 706
return status information, 707
return where object is classified, 771
Class type
analyze E071 entry, 597
Classification
change/create, 769
create with assigned values, 770
return detailed information, 772, 773
return status, 774
CLOI_PUT_SIGN_IN_FRONT, 889
CLOSE_FORM, 330
Code conversion
ISO code to SAP code, 212
ISO country code to SAP country key, 180
ISO currency code to SAP currency key, 183
SAP code to ISO code, 213
SAP country code to ISO country key, 181
Code inspector check
start, 923
Command check
checking authorization, 65
execute external command, 65
executing authorized command, 67
Commit system changes, 50
COMMIT WORK, 812, 849
COMMIT_TEXT, 330
Communication types
create address, 698
Company code
get key and name of, 709
return plant/valuation details, 914
COMPLEX_SELECTIONS_DIALOG, 252
Component
return list, 911
COMPUTE_YEARS_BETWEEN_DATES, 153
CONCATENATE, 415
Concatenate
string and currency amount, 211
Confirmation time tickets
enter, 810
Container, 937
read internal table, 939
write internal table, 940
Control commands
give to layout, 332
Control record
change for IDoc, 471
Control statement
expand/replace, 372
CONTROL_FORM, 332
Conversion exit
convert calendar quarter SAP to user-based format settings, 162
convert calendar quarter user settings to internal format calendar, 161
convert fiscal year from internal to external format, 158
convert fiscal year from user external to SAP internal format, 157
convert one-digit SAP to two-digit ISO language format, 160
convert two-digit ISO to one-digit SAP language field, 158
CONVERSION_EXIT_ALPHA_INPUT, 154
CONVERSION_EXIT_ALPHA_OUTPUT, 154
CONVERSION_EXIT_CUNIT_INPUT, 155
CONVERSION_EXIT_CUNIT_OUTPUT, 155
CONVERSION_EXIT_GJAHR_INPUT, 157
CONVERSION_EXIT_GJAHR_OUTPUT, 158
CONVERSION_EXIT_IDATE_INPUT, 159
CONVERSION_EXIT_IDATE_OUTPUT, 159, 161, 163
CONVERSION_EXIT_ISOLA_INPUT, 158
CONVERSION_EXIT_ISOLA_OUTPUT, 160
CONVERSION_EXIT_LDATE_OUTPUT, 159, 161
CONVERSION_EXIT_LUNIT_INPUT, 156

CONVERSION_EXIT_LUNIT_OUTPUT, 157
CONVERSION_EXIT_PERI5_INPUT, 161
CONVERSION_EXIT_PERI5_OUTPUT, 162
CONVERSION_EXIT_SDATE_INPUT, 162
CONVERSION_EXIT_SDATE_OUTPUT, 159, 163
CONVERSION_FACTOR_GET, 163
Convert to fraction, 178
CONVERT_ABAPSPOOLJOB_2_PDF, 165
CONVERT_DATE_TO_EXTERNAL, 160, 167
CONVERT_ITF_TO_HTML, 168
CONVERT_OTF, 172
CONVERT_OTF_MEMORY, 333
CONVERT_OTFSPOOLJOB_2_PDF, 174
CONVERT_TIME_INPUT, 175
CONVERT_TO_FOREIGN_CURRENCY, 176, 179
CONVERT_TO_FRACTION, 178
CONVERT_TO_LOCAL_CURRENCY, 179
CO-PA, 718
  read planning data, 719
  return planning data from, 720
COPO_POPUP_TO_DISPLAY_TEXTLIST, 549
COPY_TEXTS, 334
Costing-Based Controlling-Profitability Analysis, 718
COUNTRY_CODE_ISO_TO_SAP, 180
COUNTRY_CODE_SAP_TO_ISO, 181
Country-specific address, 496
CREATE_TEXT, 335
CS_BOM_EXPL_EQU_V2, 889
CS_BOM_EXPL_MAT_V2, 892
CS_WHERE_USED_MAT, 896
CSAP_MAT_BOM_READ, 897
CSEP_MAT_BOM_SELECT, 899
CSEP_MAT_BOM_SELECT_CHANGE, 900
CSEP_MAT_BOM_SELECT_WHERE_USED, 900
Currency
  convert foreign to local, 179
  convert local to foreign, 176
  convert to external format, 721
  convert to internal format, 721
Currency code
  currency not conforming to ISO, 183
Currency code (Cont.)
  flagging in TCURC, 183
Currency conversion
  alternative exchange rates, 176
  convert amounts/currency to string format, 211
  determine exchange rate from TCURR, 219
  foreign currency keys, 176
  foreign currency to local currency, 179
  from SAP system to external BAPI, 182
  from SAP system to external IDoc, 182
  local currency to foreign currency, 176
  translate SAP internal currency code to ISO currency code, 184
Currency rates
  amounts, 148
  currency keys, 148
  find by date, 148
CURRENCY_AMOUNT_SAP_TO_BAPI, 182
CURRENCY_AMOUNT_SAP_TO_IDOC, 182
CURRENCY_CODE_ISO_TO_SAP, 183
CURRENCY_CODE_SAP_TO_ISO, 184
Current stock
  select, 765
Custom help, 123
Customer
  address read, 724
  billing document, cancel, 686
  change data, 722
  change/delete inquiry, 724
  change/delete quotations, 727
  create new, 723
  display sales order, 844
  get account balance from date, 683
  inquiry, 724
  read address, 724
Customer account
  display noted items, 684
  display open items, 684
Customer balance
  display for fiscal year, 685
Customer quotation
  change/delete, 727
  create, 813

D

Data
download to PC in Excel format, 439
format for Excel, 423
import from Excel to SAP internal table, 437
import to SAP from Excel, 423
maintain across tables, 142
read from archive info system, 406
transfer from legacy to SAP, 250
transfer to PC from Excel, 423
Data archiving
create new archive file, 398
customize settings for archiving program, 383
delete data in archive database, 393
delete table data, 383
end data archiving, 383
get current handle information, 384
get next data object, 386
get next record, 387
immediately retrieve data, 386
open archives that are to be read, 397
open for write, 398
pass data record to data object, 400
put data record in current data object, 401
read data from RECORD_STRUCTURE, 390
read data object records sequentially, 387
return delete program values, 383
return handle from archive file for further processing, 402
return values from RECORD_STRUCTURE, 390
transfer statistical data, 392
write data object in archive file, 404
write statistical data to standard log, 405
write, delete, read and reload programs, 392
write, delete, reload, and read programs, 389, 405
write/reload data objects in a new archive file, 395
writing new data object, 393
Data conversion
internal table data to Excel format, 443
Data element
definition, 86
read, 86
Data object
read for archiving object, 402
read records sequentially, 387
write in archive file, 404
write to archive, 393
write/reload to new archive file, 395
Data record
pass to data object, 400
put in current data object, 401
read from current data object, 390
Database
delete log, 634
find logs, 637
load log entries, 635
Database table
return all fields and information, 912
DataProvider object, 253, 255
create, 255
send table info to client, 253
transfer from client to ABAP/4 program, 255
Date
calculate, 210
calculate difference, 153
check validity, 185
check, subtract days/months/years from date, 223
convert external to internal, 162
convert external to internal format, 159
convert from external to internal format, 186
convert internal to external, 159
convert internal to external format, 161, 163
determine new, 217
factory calendar, 187
Date check
add/subtract calendar days, 209
calculate an employment period, 210
calculate duration of days between two points, 209
calculate future calendar date, 189
calculate input parameter date, 188
calculate past/future calendar date, 189
calculate the difference between two dates, 208
calculate working date through date/number of working days, 243
check time validity, 237

Date check (Cont.)
check working days, 185
checking for corresponding dates, 186
convert dates from external to internal format, 186
determine all holidays/non-working days, 207
determine date and time of day in a duration, 198
determine date difference in seconds, 233
determine dates between two calendars, 202
determine day name/number, 220
determine days within a certain time frame, 193
determine end date and time, 199
determine first and last day of accounting period, 218
determine last day of a month, 224
determine start date/time, 231
determine the first calendar date of a period, 203
determine the new date by adding number of months, 217
determine time difference between two dates/times, 230
factory specific calendar, 201
holidays, 206
obtaining attributes of calendar days, 191
return current/previous fiscal year/month, 205
return day of week in specified language, 244
return defined number of workings days in a week, 245
return to first day of week, 244
subtract time from a provided date, 232
Date conversion
convert an external date to internal, 168
convert an internal date to external date, 167
DATE_CHECK_PLAUSIBILITY, 185
DATE_CHECK_WORKINGDAY, 185, 187
DATE_COMPUTE_DAY , 186
DATE_CONV_EXT_TO_INT , 186
DATE_CONVERT_TO_FACTORYDATE, 185, 187
DATE_GET_WEEK, 188, 244
DATE_IN_FUTURE , 189
DATE_TO_PERIOD_CONVERT , 190
Dates
determine days between, 193
find difference between, 202
Day
determine name/number, 220
number, 192
DAY_ATTRIBUTES_GET , 191
DAY_IN_WEEK, 192
DAYS_BETWEEN_TWO_DATES , 193
DDIF_DOMA_GET, 85
DDIF_DTEL_GET, 86
DDIF_FIELDINFO_GET, 87
DDIF_NAMETAB_GET, 89
DDIF_TABL_ACTIVATE, 92
DDIF_TABL_GET, 92
DDIF_TABL_PUT, 94
DDIF_VIEW_GET, 95
Decentralized system
delivery verification, 746
verify outbound delivery, 775
Decimal
determine, 182
determine for currency, 911
round number, 222, 237
Decimal point
get number of places around, 414
DECIMAL_PLACES_GET, 911
Default printer settings, 528
Default variant
read, 291
Delete program, 383
DELETE_TEXT, 336
DELETE_TEXT_FROM_CATALOG, 336
Deletion flag
cancel, 754
Delivery document
create for individual stock transport order, 779
create for sales order/item, 778
create for stock transport order, 779
Delivery item, 775
Delivery split, 779
DEQUEUE_ALL, 96
DEQUEUE_E_TABLE, 97
DEQUEUE_ES_EDIDOCE, 471
DEQUEUE_ESFUNCTION, 901, 902

Determine system of unit, 230
Development system
program deletion, 58
Device type
convert OTF to PDF format, 370
executeSmart Form, 370
return, 370
return device type of given language, 370
return Smart Form name, 370
send data to presentation server, 370
show file save dialog box, 370
Dialog box
adjust column width, 114
create custom, 563
create list of options, 580
delete variants, 315
diagnosis message plus response option, 582
display list of report variants, 311
generate message with no list formatted, 577
generate when no page created, 577
send and include specified fields to interface table., 569, 572, 574, 576
send/display fields listed in interface-table FIELDS, 570
show collected messages, 559
validate/check values in input fields, 574
Digital clock
display, 550
Dimension key
get, 197
get with text, 197
DIMENSION_CHECK, 195, 196, 197
DIMENSION_GET, 195, 196, 230, 242
DIMENSION_GET_FOR_UNIT, 195, 197, 242
Dimensions
check dimensions of a unit, 195
determine dimension keys based on unit exponents, 196
seven base units/dimensions, 196
Display profile
display message in full screen, 641
display one log entry, 642
return in tree, 640
Display variant
check existence in DB, 271, 292
Distribution list
create new, 446
return all information, 447
Document
create attachment for, 443
create in specific folder, 449
post in accounting, 665
post to accounting, 662
provide information, 450
rename, 364
send as email, 452
send existing, 458
send existing attachment, 452
send new, 452, 454
send new/uncreated, 456
Document check
billing for SD, 662
general ledger, 662
invoice receipt, 662
Document maintenance
display text in popup, 564, 565
Document tables
initialize, 812
Documentation header table, 549
Domain
read, 85
DP_CONTROL_ASSIGN_TABLE, 253, 255
DP_CREATE, 255
DP_GET_FIELDS_FROM_TABLE, 254, 255, 256
Dummy command, 71
Dummy name, 364
DURATION_DETERMINE, 198
Dynamic message, 566
Dynamic selection
convert from ranges to where format, 267
create screen, 260
enter, 260
initialize capture, 263
DYNP_GET_STEPL, 257
DYNP_VALUE_READ, 257
DYNP_VALUE_UPDATE, 257
DYNP_VALUES_READ, 257, 303
DYNP_VALUES_UPDATE, 259, 304

Dynpro
  date conversion, 167, 168
  F1 help, 127

E

E071 entry
  analyze class type, 597
EDI, 38, 463
EDI_CHANGE_CONTROL_RECORD, 471
EDI_CHANGE_DATA_SEGMENTS, 472
EDI_DOCUMENT_CLOSE_EDIT, 473
EDI_DOCUMENT_CLOSE_PROCESS, 473
EDI_DOCUMENT_DEQUEUE_LATER, 474
EDI_DOCUMENT_OPEN_FOR_EDIT, 475
EDI_DOCUMENT_OPEN_FOR_PROCESS, 476
EDI_DOCUMENT_STATUS_SET, 477
EDI_STATUS_ADD_BLOCK, 478
EDIDD, 484
EDIT_TEXT, 338
EDIT_TEXT_INLINE, 340
Editor control
  display TEXTNOTE, 588
Email
  send to user from spool list, 442
Embedding, 423
Employee ID
  get from user name, 865
Encrypted password, 53
End date
  determine, 199
END_FORM, 343
END_TIME_DETERMINE, 199
ENQUE_SLEEP, 904
ENQUEUE_E_TABLE, 98
ENQUEUE_ESFUNCT, 901
ENQUEUE_ESFUNCTION, 902
  release lock, 901
Entry
  write in SAP system log, 60
Environment variable, 72
Equipment
  change details, 730
  create, 731
  explode BOM, 889
ERGO_TEXT_SHOW, 549
Error
  return text, 347
ESFUNCTION
  request lock, 902
EVAL_FORMULA, 200
Event
  trigger for business object, 938
Event ID
  background systems, 48
Excel
  analyze sheet data, 437
  start modally, 440
  start via ASCII data, 440
EXCEL_OLE_STANDARD_DAT, 423
Exchange rate
  determine, 219
EXCHANGE_ITF, 343
EXECUTE_WINWORD, 425
Execution of Program After, 593
EXPORT_TEXT, 344
Extended Export of Lists -> see XXL, 460
Extended maintenance
  levels, 132
Extended table maintenance
  call, 132
  call at highest level, 134
  call at lowest level, 137
  maintain individual entries, 140
  maintain objects, 142
External command
  check user authorization for execution, 67
  check user authorization to run command, 71
  define, 71
  read list of, 69
External document
  reverse in accounting, 667
External operating command, 62
External system
  create job for background processing, 869

F

F1 help, 127
  display documentation data, 549
  Dynpro, 127

F1 help (Cont.)
- *popup dialog box*, 549
- *popups, get user value help*, 576

F4 dialog, 113
F4 help, 242
- *define parameter/fields of BAPI*, 741
- *directory files in application server*, 552
- *display digital clock*, 550
- *display values*, 272
- *field values*, 257
- *file manager support*, 551
- *list all tables withcheckboxes*, 580
- *locate file name*, 438
- *popups, get user value help*, 576
- *units of a dimension*, 242
- *work center*, 303

F4 values, 272
F4_CLOCK, 550
F4_DATE, 101
F4_FILENAME, 551
F4_FILENAME_SERVER, 552
F4IF_DETERMINE_SEARCHHELP, 103
F4IF_FIELD_VALUE_REQUEST, 104
F4IF_GET_SHLP_DESCR, 107
F4IF_INT_TABLE_VALUE_REQUEST, 108
F4IF_SELECT_VALUES, 111
F4IF_SHLP_EXIT_EXAMPLE, 112
F4IF_START_VALUE_REQUEST, 113
F4UT_OPTIMIZE_COLWIDTH, 114
F4UT_PARAMETER_ALLOCATE, 115
F4UT_PARAMETER_RESULT_PUT, 116
F4UT_PARAMETER_SORT, 117
F4UT_PARAMETER_VALUE_GET, 118
F4UT_RESULTS_MAP, 120
Factory calendar
- *calculate working date*, 243
- *display*, 101
- *find date*, 187
- *return date*, 201
- *return working day*, 245

Factory date, 201
- *convert*, 187

FACTORYDATE_CONVERT_TO_DATE, 201
Field
- *find assigned search help*, 103
- *set max number in popup*, 571

Field catalog
- *create for ALV FM*, 269
- *create from dictionary structure/internal table*, 276
- *outputting tables in AVL format*, 277
- *transfer from SLIS to LVC*, 269

Field contents
- *convert external to internal format*, 225

Field name
- *return current values*, 303

Field selection
- *filtering conditions*, 488

FIELDS interface table, 569
File
- *check authorization to access with key word*, 501
- *upload from client presentation*, 434
- *upload to ITF based on FORMAT_TYPE*, 349

File manager support
- *locate file in directory*, 551

File name
- *F4 help*, 551
- *get from user*, 428
- *locate in F4 help*, 438

File transfer
- *ASCII into MS-Excel format*, 439

File transfer protocol -> see FTP, 51
FILE_GET_NAME, 425
FILENAME_GET, 428
Filter object type
- *determine for BAPI*, 466

FIMA_DAYS_AND_MONTHS_AND_YEARS, 202
Final checks, 812
Financial Accounting module
- *accounting period*, 218

FIRST_DAY_IN_PERIOD_GET, 203
Fiscal year
- *convert external to internal format*, 157
- *convert internal to external format*, 158
- *customer current balance for*, 685
- *first calendar date*, 203
- *get from posting date/company code*, 708
- *return based on date/company code*, 205
- *return last calendar date*, 213
- *variant*, 190

FITRV_CALCULATOR, 552
Float types
converting character format to float type, 151
floating point number, 151
Floating format, 204
FLTP_CHAR_CONVERSION, 204
FLUSH_ENQUEUE, 121
FM name
execute Adobe Form, 347
return, 347
unique, 347
Form
close, 330, 343
determine output setting, 348
start, 371
Form element
output in opened form, 377
transfer lines to internal table, 358
Form printing, 325
open, 352
Format conversion
binary to XString, 226
convert OTF spool to PDF, 174
convert OTF to ASCII, 172
convert OTF to PDF, 172
dynamic selections from ranges to where format, 267
floating to character format, 204
OTF format to PDF, 234
OTF to printer format, 235
OTF to RAW format, 236
ranges to expression, 266
SAP to user display format number, 154
user display number to SAP format, 154
XString to binary, 227
FORMAT_MESSAGE, 553
FORMAT_TEXTLINES, 346
Formula
analyze syntactical correction, 200
FP_FUNCTION_MODULE_NAME, 347
FP_GET_LAST_ADS_ERRSTR, 347
FP_JOB_CLOSE, 348
FP_JOB_OPEN, 348
Free text
add to application log, 652
FREE_SELECTIONS_DIALOG, 260, 263, 266, 267
FREE_SELECTIONS_INIT, 263, 266
FREE_SELECTIONS_RANGE_2_EX, 266
FREE_SELECTIONS_RANGE_2_WHERE, 267
FTP
commands execution, 51
disconnect from FTP server, 52
encrypted password, 53
open/establish server connection, 51
pass command to server, 50
scramble source parameter, 53
FTP_COMMAND, 50
FTP_CONNECT, 51
FTP_DISCONNECT, 52
Full screen SAPscript editor, 338
Function block, 555
ARCHIVE_GET_NEXT_OBJECT, 387
Function code
set and simulate in command field, 922
Function group, 301
return information, 909
SMSG, 555
SZA0, 492
Function module
check existence/function group, 904
get details, 905, 909
give details, 907
return includes, 908
return name, 369
return names in function group, 301
FUNCTION_EXISTS, 904
FUNCTION_IMPORT_INTERFACE, 905
FUNCTION_INCLUDE_CONCATENATE, 907
FUNCTION_INCLUDE_INFO, 908
FUNCTION_INCLUDE_SPLIT, 909

G

General ledger
annual values, 685
reverse accounting posting, 670
validate business transactions, 668
GET_COMPONENT_LIST, 911
GET_CURRENT_YEAR, 205
GET_FIELDTAB, 912

GET_GLOBAL_SYMBOLS, 913
GET_INCLUDETAB, 913
GET_JOB_RUNTIME_INFO, 517
GET_PRINT_PARAMETERS, 518
GETDETAIL, 702
Global data, 913
GMT time stamp, 221
Goods movement
 code, 739
 post, 739
 reverse, 738
Gregorian calendar, 193
 display, 101
GUI
 check if ITS, 434
 check if normal Windows GUI, 434
 Internet Transaction Server, 434
GUI status
 custom, 300
 set custom, 303
 set for external selection screen, 300
GUI_DOWNLOAD, 428
GUI_DOWNLOAD module, 428
GUI_HAS_ACTIVEX, 433
GUI_IS_AVAILABLE, 434
GUI_IS_ITS, 434
GUI_UPLOAD, 434
GUI_UPLOAD module, 434

H

Handle
 get current information, 384
Handling unit
 create non-assigned, 743
 pack item, 744
 unpack, 744
Header data
 change for planned order, 781
 read log header data, 650
Header/item data
 display in ranges table, 740
Help document
 show in popup, 125
Help text
 display, 121
HELP_DOCULINES_SHOW, 121
HELP_OBJECT_SHOW, 123
HELP_OBJECT_SHOW_FOR_FIELD, 125
HELP_START, 127
Hierarchy
 build new, 308
 create, 704
 display, 309
Hit list
 change sorting, 117
 fill, 120
 sorting by characteristic, 117
 sorting by multiple parameters, 117
 transfer data, 120
Holiday, 206
 determine in period, 207
HOLIDAY_CHECK_AND_GET_INFO , 206
HOLIDAY_GET , 207
HR_HK_DIFF_BT_2_DATES , 208
HR_SEN_CALE_DAYS_DATE, 209
HR_SEN_CALE_DAYS_DURATION, 209
HR_SEN_CRULE_0100_DATE, 210
HRCM_AMOUNT_TO_STRING_CONVERT, 211
HRCM_STRING_TO_AMOUNT_CONVERT , 211
HTTP_SCRAMBLE, 51, 53

I

IBAN
 change, 745
ID
 clear/delete value, 322
 store value tables, 323
IDATE
 convert external to internal date format, 159
 convert internal to external date format, 159
IDoc, 38, 463
 change control record, 471
 change data records, 472
 change status programmatically, 477
 close after processing, 473
 close and save all changes, 473
 control record, 471
 create and save to database, 480

IDoc (Cont.)
CREATE mode, 477
generate inbound IDocs, 479
open for processing, 476
open to change content, 475
pass statuses on to database, 473
process input port type tRFC, 479
PROCESS mode, 477
replacement for asynchronous BAPIs, 467
Send IDoc posting for inbound processing, 481
transfer to ALE layer, 483
unlock, 471
unlock post-closing, 474
write one or more status records to the database, 482
IDOC_INBOUND_ASYNCHRONOUS, 479
IDOC_INBOUND_SINGLE, 479
IDOC_INBOUND_WRITE_TO_DB, 480
IDOC_START_INBOUND, 481
IDOC_STATUS_WRITE_TO_DATABASE, 482
IMPORT_TEXT, 349
Inbound delivery
verify from decentralized system, 746
Inbound IDoc
create/save in ABAP Dictionary tables, 480
INCLUDE command
expand in text lines, 373
Includes
return all, 913, 920
INIT_TEXT, 350
Inline editor
edit text, 340
Input help
elementary, 117
sorting a hit list, 117
Inspection plan
create, 748
Interest calculation method, 193
Interface, 905
Internal array buffer
import status of objects, 933
Internal order
create for transferred data, 750
Internal SAP format
convert text to user format, 154
Internal status buffer
delete objects, 927
Internal table
display data in popup, 554
display F4 help, 108
fill/display, 252
output as list, 283
output in ALV grid format, 277
output in ALV list format, 286
read from container, 939
receive, 557
replace document parameters, 566
show table entries, 289
transfer data from server to PC, 428
write in container, 940
Internal table data
convert to Excel format, 443
display in popup, 584
International Bank Account Number, 745
International System of Units, 196
Interrupt request, 404
Invoice receipt
check for availability to post to accounting, 671
post to accounting, 673
transfer to accounting, 673
IP address
search for system, 47
ISO
conversion, 180
convert code to SAP language code, 212
convert measurement, 241
country code, 180
internal country key, 181
translate currency code, 183
ITF format
replacing symbols with values, 374
ITF text
convert into desired format, 344
save to local file, 344

J

Job
background request, 526

Job (Cont.)
create for background processing from external, 869
delete, 507
delete job log, 507
find background requests, 507
get status of background processing system, 870
identify, 525
insert a background task in background request, 526
read header table, 509
read specified log and store, 512
read step list table, 509
release authorization, 521
return status, 544
select through specified criteria, 510
Job log
read specified job, 512
show log for specified job, 513
store in internal table, 512
Job scheduling
open without dialog, 525
Job step
submit to job, 526
JOB_CLOSE, 521
JOB_OPEN, 525
JOB_SUBMIT, 526

K

K_WERKS_OF_BUKRS_FIND, 914
KCD_EXCEL_OLE_TO_INT_CONVERT, 437
KD_GET_FILENAME_ON_F4, 438
Key
check, 606
Key date view
create, 748
Key word
DELETE DATASET, 501
OPEN DATASET, 501
READ DATASET, 501
TRANSFER, 501

L

Language
determine code/character, 228
field, 158
save externally, 212
Language dependent status
convert to 40 char line, 935
Language import parameter
return device type, 370
LANGUAGE_CODE_ISO_TO_SAP, 212
LANGUAGE_CODE_SAP_TO_ISO, 213
Language-dependent abbreviation, 929
convert to internal format, 934
LAST_DAY_IN_PERIOD_GET , 213
Layout set, 358
LDATE
convert internal to external date format, 161
Linear congruence method, 919
Lines
transfer between tables, 361
transfer to internal table, 358
List
append block list data to blocked ALV, 272
assign TOP-OF-PAGE to the ALV display, 275
change current information of, 283
change the current information of ALV list, 289
convert saved list object to ASCI, 268
create a hierarchical, sequential list of two tables, 283
current field catalog, 282
current information, 282
current layout, 282
current scroll state, 282
delete last saved from memory, 267
filter criteria, 282
get the current information of ALV list, 286
output internal tables in ALV list format, 286
save/convert to HTML, 324
sort sequence, 282
List import
prepare from memory, 267
List object
convert to ASCI format, 268
display Saved, 324

List object (Cont.)
display/manipulate list objects, 460
List spool request
convert to PDF, 165
List viewer, 462
LIST_FREE_MEMORY, 267
LIST_FROM_MEMORY, 267, 268
LIST_TO_ASCI, 268
Literals
evaluate, 200
LOAD_PRINT_PARAMETERS, 528
Local lock container
delete collected locks, 128
send to lock server, 121
Local memory
delete contents, 493
initialize, 624
log header data, 629
log parameters, 630
message parameters, 631
messages without parameters, 632
single message without parameters, 633
write application log data to database, 629
Lock
assign to LUW for table, 98
delete from local lock container, 128
release, 901
release for LUW, 96
request ESFUNCTION, 902
view list, 98
Log
application event, 40, 619
delete from database, 634
display in local memory, 623
find in database, 637
initialize in local memory, 624
load entries from DB, 635
read header data, 650
search in memory, 648
Log data
read for specified object, 626
save from memory to database, 636
Log handle
delete all messages from main memory, 654
Log header data
write to local memory, 629
Log memory
free, 621
Log message
delete, 654
display, 601
Log parameters
write to local memory, 630
Logical database, 390
Logical file name
assign physical file name, 425
logical file path, 425
platform-specific file name, 425
Logical receiver system
determine for asynchronous BAPI, 465
transfer to function module, 466
Logical system information, 54
Logical Unit of Work, 488
Logon system
specify name, 54
Long text
create, 335
LUW
assign lock for table, 98
release all locks, 96
release lock for TABNAME table, 97
LVC type, 269
LVC_FIELDCATALOG_MERGE, 269, 270
LVC_TRANSFER_FROM_SLIS, 269
LVC_VARIANT_DEFAULT_GET, 270
LVC_VARIANT_EXISTENCE_CHECK, 271
LVC_VARIANT_F4, 272

M

Main memory
delete all messages via I_LOG_HANDLE, 654
Maintenance table
combining into one dialog, 142
Maintenance/service notification
create new, 677
save change, 679
Maintenance/service order
change, 681
read detail data, 679
Manual cost allocation
post/reverse to accounting, 675

Master data
send/distribute changes, 469
Master production scheduling (MPS), 765
MASTER_IDOC_DISTRIBUTE, 467, 483
Material
cancel deletion, 754
change classification, 769
change reservation, 824
convert source to target unit, 216
convert UOM, 150
create classification, 769
create/change reservation, 825
current stock, 765
determine receipt quantity, 751
determine source of supply, 848
determine stock requirement, 917
edit, 754
explode BOM, 892
flag for deletion, 754
get list in logon language, 758
read/return details, 757
select requirement list, 765
unpack from HU, 744
Material BOM create, 753
Material components
add list to network, 768
Material data
create, 250
create en masse, 762
mass maintenance, 762
Material detail
get client specific material, 755
Material document
cancel, 738
create for goods movement, 739
read/display header/item data, 740
Material group
determine the source of supply, 848
Material master data, 762
change existing, 759
create new, 759
read all, 755
Material master maintenance
create batch input data, 914
Material requirements planning (MRP), 765, 781
MATERIAL_BTCI_SELECTION_NEW, 914
MATERIAL_UNIT_CONVERSION , 214
Materials Management, 216
MD_CONVERT_MATERIAL_UNIT , 216
MD_MRP_LIST_API , 915
MD_POPUP_SHOW_INTERNAL_TABLE, 554
MD_STOCK_REQUIREMENTS_LIST_API, 917
Measurement conversion
convert unit measurement with formula, 238
from business to commercial, 155
from business to technical format, 157
from commercial to business, 155
from internal to external format, 155
from internal to user external format, 157
from technical to business unit, 156
from user external to SAP internal, 155
from user external to SAP internal format, 156
ISO measurement units to SAP internal code, 241
measurement unit values, 237
SAP internal measurement to ISO code, 241
Memory
delete log, 655
prepare list import, 267
Memory buffer, read, 333
Message
add to application log, 651
check if message handler is active, 554
collect future messages, 557
configure for question to user, 578
custom, 567
delete from application log, 653
delete from local memory for object, 624
delete previously collected, 557
determine number in line interval, 555
display dialog box with previously collected messages, 559
display list in popup, 641
flag users to receive message, 586
format, 553, 585
get in internal table, 557
initialize/activate handler, 557
search in memory, 648
send custom message to user, 567
severity of errors, 560

Message (Cont.)
show diagnosis/options, 582
stop collection, 560
store in global memory, 561
write from table to message field, 589
write in local memory, 631
write multiple in local memory (no parameters), 632
Message field
T100, 589
Message handler
check if active, 554
Message text
display, 589
MESSAGE_STORE, 561
MESSAGES_ACTIVE, 554
MESSAGES_COUNT, 555
MESSAGES_GIVE, 557
MESSAGES_INITIALIZE, 557
MESSAGES_SHOW, 559
MESSAGES_STOP, 560
Module pool screens
create long text, 335
Month
choose, 582
provide name, 217
MONTH_NAMES_GET , 217
MONTH_PLUS_DETERMINE , 217
MRP
determine requirements list, 915
MS_EXCEL_OLE_STANDARD_DAT, 439
Multiple selection screen, 252, 304

N

Navigation tree
return display profile, 640
Network components
create multiple, 768
Nonassigned handling unit
create, 743
Nonworking days, 207
Number
convert to string format, 211
convert to words, 414
generate random, 919

Number (Cont.)
generate random within range, 920
move sign, 889
replace temporary with real in buffer, 932
round to decimal, 222
Number range
assign free numbers for, 409
check if number lies in interval, 406
create, change, delete intervals, 411
interval information, 408
lock, 410
lock number range, 411
unlock, 410
write changes from local memory to DB, 413
NUMBER_CHECK, 406
NUMBER_GET_INFO, 408
NUMBER_GET_NEXT, 409
NUMBER_RANGE_DEQUEUE, 410
NUMBER_RANGE_ENQUEUE, 411
NUMBER_RANGE_INTERVAL_UPDATE, 411
NUMBER_RANGE_UPDATE_CLOSE, 413
Numerator and denominator, 178
Numeric field, 889
edit/check, 918
NUMERIC_CHECK, 918

O

Object
append to transport task and lock, 594
assign free numbers, 409
check, 606
check for subobject, 628
conver output to printer, 235
convert output text to PDF, 234
convert output to RAW format, 236
create characteristics, 700
delete entries in local memory, 624
delete messages from local memory, 624
find where-used list, 306
insert in transport, 608
insert into transport, 604
maintain object and check user authorization, 607
return classes, 771

Object directory entry, 616
display parameters for sent object, 616
Object Linking and Embedding -> see OLE, 423
Object number
replace temp number with real number, 932
Object statuses
maintain/display, 928
OBJECT_NUMBER_GET_xx, 931
Objects
import status to buffer, 933
OLE, 423, 437
2.0 standard formatting, 439
Open items
display, 684
OPEN_FORM, 352
Options
restrict selection, 321
Oracle database
environment variable value, 72
Order confirmations
enter for purchase order, 809
Order detail
read data, 679
Order maintenance
object list, 681
production resources/tools, 681
Organizational unit
read all material master data, 755
OTF
convert to text format, 333
format, 234, 235, 236
OTF spool
convert to PDF, 174
Outbound delivery
verify from decentralized system, 775
OWN_LOGICAL_SYSTEM_GET, 54

P

Package details, 600
Page layout
control commands, 332
Paragraph name
convert style/form, 343
Parameter
placeholders, 566
reserve space, 115
return information, 302
PBO, 323
PDF, 234
close output job, 348
Period
convert, 190
first calendar date, 203
PERIOD_DAY_DETERMINE , 218
PF status, 322
Planned independent requirements
change, 816
create with authorization check, 818
Planned order
change header data, 781
check existence, 784
create with BOM explosion, 782
delete, 783
display F4 help, 786
get detailed list, 786
read/get details, 785
Planning data
overwrite, 720
read for CO-PA, 719
Plant
determine stock requirement, 917
return details, 914
Platform-independent logical file names, 425
Pocket calculator, 552
Popup
add custom pushbuttons, 572
bringing up documents in popup, 123
define buttons, 565
display ALV variants, 294
display any text to user, 581
display box with alternative decisions with user-defined buttons, 583
display formatted address, 499
display internal table data, 584
display pocket calulator, 552
display text created by document maintenance Transaction SE61, 564
display/select table entries, 289
question user/respond to user answer, 578

Popup (Cont.)
select month, 582
send custom message, 567
send display text document, 566
send group message for users' next log on, 586
set maximum number of fields to be displayed, 571
show any document, 125
show text list, 549
text editor, 588
user confirms processing step, 563
Popup message
send, 74
POPUP_CONTINUE_YES_NO, 563
POPUP_DISPLAY_TEXT, 564
POPUP_DISPLAY_TEXT_USER_BUTTON, 565
POPUP_DISPLAY_TEXT_WITH_PARAMS, 566
POPUP_FOR_INTERACTION, 567
POPUP_GET_VALUES, 569, 576
POPUP_GET_VALUES_DB_CHECKED, 570
POPUP_GET_VALUES_SET_MAX_FIELD, 571
POPUP_GET_VALUES_USER_BUTTONS, 572
POPUP_GET_VALUES_USER_CHECKED, 574
POPUP_GET_VALUES_USER_HELP, 576
POPUP_NO_LIST, 577
POPUP_TO_CONFIRM, 578
POPUP_TO_DECIDE_LIST, 580
POPUP_TO_DISPLAY_TEXT, 581
POPUP_TO_INFORM, 582
POPUP_TO_SELECT_MONTH, 582
POPUP_WITH_3_BUTTONS_TO_CHOOSE, 583
POPUP_WITH_TABLE_DISPLAY, 584
Post
create in accounting, 665
Posting date
determine period, 190
POV
display internal table F4 help, 108
Precommit, 812
Presentation server
open winword, 425
Pretty Printer, 54
PRETTY_PRINTER, 54
Print
get screen parameters/selection options, 298
print selection screen data, 529
report output, 529
store parameters, 545
Print parameters
define, 518
display, 518
display for specified key, 528
modify, 518
Print set
change layout, 371
combining layouts, 371
PRINT_SELECTIONS, 529
PRINT_TEXT, 355
PRINT_TEXT_ITF, 357
Printer
format, 235
loading parameters, 528, 545
Printer output
combine layouts, 371
Printing
close form, 343
Process on Value Request -> see POV, 108
Production order
change header data, 807
create, 808
enter conformation time tickets for operations, 810
enter order conformation, 809
Production Planning, 781
Profile
return in standard application log, 643
Profile assignment
change for user, 864
delete all for user, 865
Profitability Analysis, 720
Program
components list for structure, 911
delay processing for specified time, 922
delay processing in seconds, 904
delete, 58
display internal table data in popup, 554
display package information, 600
find background job requests, 507
MDBAPI01, 782

Program (Cont.)
read status/job log, 512
return all includes, 920
return includes used, 913
RSARCH07, 396
RSSCA1T0, 160
RSSLG150, 622
RSSPO430, 575
RSSPO450, 572
SBAL_DEMO_03, 650
SBAL_DEMO_04_POPUP, 641
SBAL_DEMO_04_SINGLE_LOG, 643
SBAL_DEMO_04_STANDARD, 643
SBAL_DEMO_05, 635
SBOOKA, 399
SBOOKD, 383
schedule for background processing, 514
submit, 318
Progress indicator
display, 318
Project definition
set/reset status, 871
Purchase order
cancel/revoke, 801
change, 786
create, 791
release/approve order, 799
return detailed information, 795
return list of items, 798
Purchase requisition
change, 802
change with authorizations, 819
create, 804, 815
create with authorizations, 820
delete/close, 822
release/approve by item, 822
Purchase requisition item
release/approve, 822
Purchasing contract
create, 712
get details, 715
make changes to, 709

Q

QF05_RANDOM, 919
QF05_RANDOM_INTEGER, 920
Quant data
return, 868
Quantity
convert to different UOM, 150
convert UOM, 214
Question and response, 563
Quotation
create for customer, 813

R

Ranges
fill/display internal table, 252
Ranges format
convert to expression, 266
RAW format, 236
READ_EXCHANGE_RATE, 219
READ_FORM_ELEMENTS, 358
READ_FORM_LINES, 358
READ_TEXT, 169, 359
READ_TEXT_INLINE, 361
Receipt quantity
determine, 751
Receiver
determine for synchronous BAPI, 468
determine for synchronous BAPI from ALE, 466
generate, 467
RECORD_TAB
reserve space in parameter, 115
REFER_TEXT, 363
Registered application servers
provide list to caller, 76
Remote function
starting new task from non-SAP systems, 45
RENAME_TEXT, 364
Report
assign variant values to parameter/selection options, 306
return current selection to SELECTION_TABLE, 301

Report (Cont.)
return info for parameters, selection, default values, 302
return list in box, 311
variance existance, 316
Report output
print selection screen data, 529
Report variants
return list to dialog box, 311
Repository browser
screen details, 296
Requirement coverage requests create, 815
Requirement list
select for material, 765
Reservation
change, 824
create for material, 825
details, 826
RESET_ENQUEUE, 128
REUSE_ALV_BLOCK_LIST_APPEND, 272
REUSE_ALV_BLOCK_LIST_DISPLAY, 274
REUSE_ALV_BLOCK_LIST_INIT, 272, 274
REUSE_ALV_COMMENTARY_WRITE, 275, 278
REUSE_ALV_FIELDCATALOG_MERGE, 270, 276, 277, 286
REUSE_ALV_GRID_DISPLAY, 277, 286
REUSE_ALV_GRID_LAYOUT_INFO_GET, 282
REUSE_ALV_GRID_LAYOUT_INFO_SET, 283
REUSE_ALV_HIERSEQ_LIST_DISPLAY, 283
REUSE_ALV_LIST_DISPLAY, 286
REUSE_ALV_LIST_LAYOUT_INFO_GET, 286
REUSE_ALV_LIST_LAYOUT_INFO_SET, 289
REUSE_ALV_POPUP_TO_SELECT, 289
REUSE_ALV_VARIANT_DEFAULT_GET, 291
REUSE_ALV_VARIANT_EXISTENCE, 292
REUSE_ALV_VARIANT_F4, 293
REUSE_ALV_VARIANT_SELECT, 294
RFC
check destination, 49
checking user authorization, 47
close open connection, 55
command check, 65
command execute, 67
destination, 49
RFC (Cont.)
tRFC, 38, 463
RFC_CONNECTION_CLOSE, 55
RFC_READ_TABLE, 56
RH_GET_DATE_DAYNAME , 220
RH_START_EXCEL_WITH_DATA, 440
RKD_WORD_WRAP, 413
RKE_TIMESTAMP_CONVERT_INPUT, 221
RKE_TIMESTAMP_CONVERT_OUTPUT , 222
ROLLBACK WORK, 850
ROUND, 222
Routing
create with subobjects, 827
view complete history, 827
RP_CALC_DATE_IN_INTERVAL , 223
RP_LAST_DAY_OF_MONTHS, 224
RPY_DYNPRO_READ, 296
RPY_MESSAGE_COMPOSE, 585
RPY_TRANSACTION_READ, 297
RS_ACCESS_PERMISSION, 296
RS_CONV_EX_2_IN, 225
RS_COVERPAGE_SELECTIONS, 298
RS_CREATE_VARIANT, 299, 301
RS_DELETE_PROGRAM, 58
RS_EXTERNAL_SELSCREEN_STATUS, 300, 304, 305
RS_FUNCTION_POOL_CONTENTS, 301
RS_GET_ALL_INCLUDES, 920
RS_REFRESH_FROM_SELECTOPTIONS, 301
RS_REPORTSELECTIONS_INFO, 299, 301, 302
RS_SELECTIONSCREEN_READ, 303, 304
RS_SELECTIONSCREEN_UPDATE, 304
RS_SEND_MAIL_FOR_SPOOLLIST, 442
RS_SET_SELSCREEN_STATUS, 300, 303, 305
RS_SUBMIT_INFO, 303, 305
RS_SUPPORT_SELECTIONS, 306
RS_TOOL_ACCESS, 306
RS_TREE_CONSTRUCT, 308, 309
RS_TREE_LIST_DISPLAY, 308, 309
RS_VARIANT_CATALOG, 311
RS_VARIANT_CHANGE, 313
RS_VARIANT_CONTENTS, 298, 314
RS_VARIANT_DELETE, 315
RS_VARIANT_EXISTS, 316

RS_VARIANT_VALUES_TECH_DATA, 311, 317
RSLG_WRITE_SYSLOG_ENTRY, 60
RSPO_DISPLAY_SPOOLJOB, 530
RSPO_DOWNLOAD_SPOOLJOB, 531
RSPO_FIND_SPOOL_REQUESTS, 531
RSPO_OUTPUT_SPOOL_REQUEST, 533
RSPO_RETURN_ABAP_SPOOLJOB, 535
RSPO_RID_SPOOLREQ_LIST, 536
RSTS_CLOSE, 536
RSTS_GET_ATTRIBUTES, 537
RSTS_OPEN_RLC, 539
RSTS_READ, 540
Runtime
  control commands for page layout, 332
  information read, 87, 89
RV_ORDER_FLOW_INFORMATION, 921
RZL_SLEEP, 922
RZL_SUBMIT, 318

S

Sales document
  read flow after delivery and billing, 921
Sales order
  change/delete, 839
  create, 841
  create delivery document, 778
  display for customer, 844
  simulate, 845
SAP
  call browser with specified URL to frontend, 875
  environment variable, 72
  internal country key, 180
SAP country code
  sonvert, 181
SAP GUI, 433
  check availability, 434
SAP Help
  download files, 46
SAP internal currency code
  translate to ISO, 184
SAP internal table
  import data from Excel, 423
SAP language code
  convert to ISO, 213
SAP print parameter storage, 545
SAP system log
  write entry in, 60
SAP_CONVERT_TO_XLS_FORMAT, 443
SAPGUI_PROGRESS_INDICATOR, 318
SAPGUI_SET_FUNCTIONCODE, 922
SAP-internal measurement, 241
SAPoffice user
  return all information, 459
SAPscript
  assign value to a symbol, 376
  convert text to HTML, 168
  create text module, 335
  create texts as reference, 363
  format line length, 346
  output to buffer, 333
  print text in ITF, 357
  read text module, 359
SAPscript editor
  call full screen, 338
  exact print representation, 357
SAVE_TEXT, 365
Scheduling agreement
  change details, 829
  create, 832
  get details, 836
Scheduling agreement delivery
  create/change schedule lines, 847
SCI_INSPECT_LIST, 923
SCMS_BINARY_TO_XSTRING, 226, 227
SCMS_XSTRING_TO_BINARY, 226, 227
SCP_CODEPAGE_FOR_LANGUAGE, 228
SCP_REPLACE_STRANGE_CHARS, 228
Screen elements, 296
  flow logic, 296
  screen exits for the repository browser, 296
Screen field
  change contents without PBO, 259
  find F4 help, 104
Screen field values
  read before PAI value transport, 257
Screen sequences
  run without user input, 922

Script conversion
display script on web, 168
Scrolling, 319
SCROLLING_IN_TABLE, 319
SD_DATETIME_DIFFERENCE , 230
SDATE
convert external to internal date format, 162
convert internal to external date format, 163
Search help
determine, 103
distinguishing parameters, 118
elementary, 114
exit, 112
find current parameter contents, 118
hit list, 120
hit list display, 114
interface and design, 112
return complete description, 107
select value, 111
selection method additional fields, 115
start F4 dialog, 113
template, 112
Search help parameter
changing a result, 116
name, 115
SELECT_OPTIONS_RESTRICT, 321
SELECT_TEXT, 367
Selection options
restrict, 321
Selection screen
check for changed parameters, 314
create variant, 299
GUI status for external screen, 300
load default values, 306
print data, 529
restrict options, 321
return parameters/selection options, 298
return report contents to table, 301
set custom GUI status, 303
technical values, 317
update values to fields, 304
Self-programmed help, 125
Service entry sheets
create, 729
SET_PRINT_PARAMETERS, 542
SHOW_JOBSTATE, 544
SI unit, 230
SI_UNIT_GET, 230
Smart Form
convert to PDF, 172
execute, 369
process written logic, 369
read errors/warnings, 371
return device type for a given language, 370
SMD tool, 469
SMSG, 555
SO_ATTACHMENT_INSERT_API1, 443
SO_ATTACHMENT_READ_API1, 445
SO_DLI_INSERT_API1, 446
SO_DLI_READ_API1, 447
SO_DOCUMENT_INSERT_API1, 449
SO_DOCUMENT_READ_API1, 450
SO_DOCUMENT_SEND_API1, 452, 454
SO_EXPRESS_FLAG_SET, 586
SO_NEW_DOCUMENT_ATT_SEND_API1, 454
SO_NEW_DOCUMENT_SEND_API1, 456
SO_OLD_DOCUMENT_SEND_API1, 458
SO_USER_READ_API1, 459
Source of supply, 848
Source unit
convert to target, 216
Special characters
replace, 228
SPELL_AMOUNT, 414
Spool, 505
create output request, 533
download data to presentation server, 531
set archive parameters, 542
set print parameters, 542
TemSe, 537
Spool job
display details, 530
open, 348
Spool list
create/send email, 442
Spool request
display list with IDs, 536
display output, 535
display using ALV, 536
find requests, 531
return data to internal table, 531

SSF_FUNCTION_MODULE_NAME, 369
SSF_GET_DEVICE_TYPE, 370
SSF_READ_ERRORS, 371
Start specifications, 514
START_FORM, 371
START_TIME_DETERMINE, 231
Statistical data
  transfer from ABAP archiving to DB, 392
  transfer from DB to ABAP archiving, 389
  write to standard log, 405
Status
  check if set, 926
  delete object/record from buffer, 927
Status buffer
  replace temp with real number, 932
Status management buffer
  delete, 924
Status object
  generate, 931
  read, 933
Status record
  create, 478
  write one or more to database, 482
  write to change pointer, 470
STATUS_BUFFER_REFRESH, 924
STATUS_CHANGE_EXTERN, 924, 933
STATUS_CHANGE_INTERN, 925
STATUS_CHECK, 926
STATUS_DELETE_FROM_BUFFER, 927
STATUS_MAINTAIN, 928
STATUS_NUMBER_CONVERSION, 929, 934
STATUS_OBJECT_CREATE, 931, 932
STATUS_OBJECT_SWITCH_NUMBER, 931, 932
STATUS_PRE_READ, 933
STATUS_READ, 933
STATUS_TEXT_CONVERSION, 929, 934
STATUS_TEXT_EDIT, 924, 935
Step loop, 257
  define line, 257
STEPL field, 257
Stock requirements
  determine list, 917
STORE_PRINT_PARAMETERS, 545
String
  adjust separators/cases, 417
String (Cont.)
  concatenate, 211, 415
  format, 211
  input string in reverse order, 416
  multibyte code page, 415
  split the text, 418
  split to substrings, 418
String amount
  convert to actual, 211
STRING_CONCATENATE, 415
STRING_REVERSE, 416
STRING_UPPER_LOWER_CASE, 417
SUBMIT, 305
  get mode information, 305
SUBTRACT_TIME_FROM_DATE , 232
Subtree
  insert, 308
SUSR_USER_AUTH_FOR_OBJ_GET, 502
SWA_STRING_SPLIT, 418
SWC_ELEMENT_GET, 937
SWC_ELEMENT_SET, 937
SWC_TABLE_GET, 939
SWC_TABLE_SET, 940
SWE_EVENT_CREATE, 938
SWI_DURATION_DETERMINE , 233
SWO_CREATE, 940
SWO_FREE, 940, 942
SWO_INVOKE, 940, 941
SX_OBJECT_CONVERT_OTF_PDF, 234
SX_OBJECT_CONVERT_OTF_PRT , 235
SX_OBJECT_CONVERT_OTF_RAW , 236
SXPG_CALL_SYSTEM, 62
SXPG_COMMAND_CHECK, 65
SXPG_COMMAND_EXECUTE, 67
SXPG_COMMAND_LIST_GET, 69
SXPG_DUMMY_COMMAND_CHECK, 71
Symbol
  replace in text lines, 374
System of Unit, 230
System status
  set/delete, 925

T

Table
  analyze structure, 256

Table (Cont.)
create dynamically from table, 92
create from text headers, 367
create/activated in ABAP Data Dictionary, 92
determine text elements, 358
DOKHL, 549
edit for change document, 881
FIELDS, 570
FIELDVALUES, 303
generate IDoc when changed, 469
lock, 121
maintenance PF status, 322
read from ABAP Data Dictionary, 92
read from external server, 56
read text, 87
SELECTION_TABLE, 301
send information to client, 253
SOFFCONT1, 449
SPOOLREQUESTS, 531
T002, 328
T005, 180
T006A, 157
T100, 589
TCURR, 148
TFACD, 201
TSAC(T), 491
TTXID, 327
TTXOB, 329
VALUES, 323
write information, 94
Table definition
find, 92
Table structures
scrolling, 319
Table text
display in popup dialog box, 549
TCP-based network, 52
Technical values of selection screen fields, 317
TemSe, 512
close object, 536
temporary sequential data, 537
TemSe object
edit, 539
get attributes, 537
TemSe object (Cont.)
open to read line by line conversion, 539
read data, 540
TERM_CONTROL_EDIT, 588
Text
change/create, 365
check language definition, 328
copy from text catalog, 334
delete from catalog, 336
display in popup, 564, 565
edit in inline editor, 340
edit lines, 338
expand/replace control statements, 372
read from text file/memory, 359, 361
read in defined work area, 361
return text about error, 347
saving references, 363
show custom text to user, 581
simple characters in, 228
split, 418
word wrap, 413
Text creation text
transfer to update, 330
update to log files, 330
Text editor
create in popup, 588
Text elements
determine in table layout, 358
Text ID
check for definition, 327
Text lines
defined formatting attributes, 379
display/change text lines, 361
expand INCLUDE commands, 373
naming empty fields, 379
output to table LINES, 379
print as they appear, 357
replacing symbols with values, 374
Text module
check user authorization, 327
check valid characters, 329
creating a table from text headers, 367
delete in a text file, 336
prepare for output device, 355
reference change, 363
replacing dummy names, 364

Text module (Cont.)
update and commit, 330
write to text file/memory, 365
Text object
check validity/definition, 329
Text symbols
changing the value, 376
TEXT_COMMAND_REPLACE, 169
TEXT_CONTROL_REPLACE, 372
TEXT_INCLUDE_REPLACE, 169, 373
TEXT_SPLIT, 418
TEXT_SYMBOL_REPLACE, 169, 374
TEXT_SYMBOL_SETVALUE, 376
TEXTNOTE, 588
TH_DELETE_USER, 71
TH_ENVIRONMENT, 72
TH_GET_OWN_WP_NO, 73
TH_POPUP, 74
TH_REMOTE_TRANSACTION, 75
TH_SERVER_LIST, 76
TH_STOP_WP, 77
TH_USER_INFO, 78
TH_USER_LIST, 79
Time
add with date correction, 148
add/subtract to date, 147
check validity, 237
convert from external to internal format, 175
determine length between start/end, 198
difference between date/time, 230
subtract from date, 232
update, 148
Time confirmations
maintain for operations, 676
Time conversion
from external to internal SAP format, 175
GMT to local date/time, 222
local time/date to GMT, 221
Time tickets
enter for confirmation, 810
TIME_CHECK_PLAUSIBILITY, 237
TOP-OF-PAGE event, 275
TR_APPEND_LOG, 593
display memory messages, 601
TR_APPEND_TO_COMM_OBJS_KEYS, 594
TR_CHECK_TYPE, 597
TR_DEVCLASS_GET, 600
TR_FLUSH_LOG, 601
TR_INSERT_NEW_COMM, 601
TR_OBJECT_CHECK, 606
TR_OBJECT_INSERT, 604
TR_OBJECTS_CHECK, 607
TR_OBJECTS_INSERT, 608
TR_READ_COMM, 610
TR_SYS_PARAMS, 612
TR_TADIR_INTERFACE, 613
TR_TADIR_POPUP_ENTRY_E071, 616
Transaction
calling code in same/separate task, 45
check authorization for execution, 502
CS03, 897
execute remotely, 75
MD04, 917
MD06, 915
MDBA, 781, 782, 783
ME21N, 791, 795
ME22N, 786
ML81N, 729
MM01, 914
MM02, 754
MM42, 754
SE38, 593
SE61, 549, 564
SE74, 343
SE92, 60
SLG1, 637, 643
SM02, 74
SM12, 98
SM21, 60
SM36, 514
SM37, 508
SM58, 479
SO10, 327
SP01), 530, 535
VL09, 555
XK01, 866
Transaction code
get details, 297
Translate currency, 176
Translation rate, 148
Transport
append object to task, 594

Transport (Cont.)
change object directory entry, 613
changing object, 607
check object/key, 606
create object directory entry, 613
delete object directory entry, 613
determine global system parameters, 612
insert object and key, 604
insert objects and keys, 608
lock object, 594
read all request data, 610
read object directory entry, 613
Transport Management System, 591
Transport Organizer
create new transport request or task, 601
object tool, 607, 608
Transport request
create new, 601
Tree structure
JavaScript, 46
Netscape, 46
tRFC, 463
return information for receiver, 80
TRFC_RECEIVER_INFO, 80
TXW_TEXTNOTE_EDIT, 588
Type
read text, 87
return text information, 87

U

Unit
check dimensions, 195
convert measurement, 238
search, 240
Unit check
two units with same dimensions, 239
Unit conversion
converting quantities from one unit to another, 214
converting source unit to target unit, 216
Unit of measure
convert external to internal, 155, 156
convert internal to external, 155, 157
convert quantity, 214
convert quantity measurements, 150
Unit of measure (Cont.)
convert with the measurements unit table, 163
round decimals in unit values, 163
UNIT_CONVERSION_SIMPLE, 237
UNIT_CONVERSION_WITH_FACTOR, 238
UNIT_CORRESPONDENCE_CHECK, 239
UNIT_GET , 240
UNIT_OF_MEASURE_ISO_TO_SAP, 241
UNIT_OF_MEASURE_SAP_TO_ISO, 241
UNIT_OF_MEASUREMENT_HELP, 242
User
authorization, 502
change attributes, 852
change profile assignment, 864
change role assignment, 850
change/create, 854
check authorization for specific module, 327
check existence, 860
check existence of specified, 81
check if authorized to edit object, 607
create in all systems, 857
delete, 859
delete profile assignment, 865
delete role assignment, 851
get details, 860
list of all logged in users, 79
lock out of system, 863
log off, 71
prompt to choose option, 580
provide details, 78
return list based on selection criteria, 862
select group to receive message, 586
send popup message to, 74
show dynamic message to, 566
transfer roles/authorizations during upgrade, 593
unlock, 864
User display
convert to SAP format, 154
User name
get employee ID, 865
User status
change, 935
set/delete, 924
USER_EXISTS, 81

V

Validate business transactions, 668
Valuation area
return details, 914
Value
assign to SAPscript, 376
convert measurement, 237
supply to an standard selection field, 304
Variable
evaluate, 200
store content, 545
Variant
change parameters, 313
check existence for report, 316
create for selection screen, 299
delete, 315
return value for existing, 314
select options, 313
selection dialog box, 293
Variant value
assign to parameter, 306
Vendor
create/change details, 866
get detailed data, 867
mark for deletion, 867
View
defined parameter details, 131
get information to change data, 131
maintenance control block table, 131
read, 95
View cluster maintenance
call, 142
View structure
lock/unlock data, 130
VIEW_AUTHORITY_CHECK, 128
VIEW_ENQUEUE, 130
VIEW_GET_DDIC_INFO, 131
VIEW_MAINTENANCE, 132
VIEW_MAINTENANCE_CALL, 134
VIEW_MAINTENANCE_LOW_LEVEL, 137
VIEW_MAINTENANCE_SINGLE_ENTRY, 140
VIEW_SET_PF_STATUS, 322
VIEWCLUSTER_MAINTENANCE_CALL, 142
VRM_DELETE_VALUES, 322
VRM_SET_VALUES, 322, 323

W

Warehouse
detailed quant data, 868
WDKAL_DATE_ADD_FKDAYS , 243
Web browser
call with URL, 875
Week
calculate based on date, 188
first day of, 244
return day for given language, 244
return day number, 192
WEEK_GET_FIRST_DAY, 244
WEEKDAY_GET, 244
Weekend
process further logic, 192
Where-used list
find, 306
find for BOM, 896
find material in BOM, 900
Winword file
open from presentation server, 425
WLB3_GET_NUMBER_OF_WORKDAYS, 245
Word wrap, 413
Work area
clear text header/line table, 350
initialize text header/line table, 350
read text, 361
Work process
get details, 73
stop/delete session, 77
Working date, 243
Working day, 245
Wrapper function module, 454
WRITE_FORM, 377
WRITE_FORM_LINES, 379
WRITE_LIST, 324
WRITE_MESSAGE, 589
WWW_LIST_TO_HTML, 324

X

XMSG, 593
XPRA, 593
  display messages from memory, 593
  log output, 593
  read and display data, 593
  use Transaction SE38 to write to memory, 593
XString format
  convert to binary, 227
XXL
  transfer data list, 460
  transfer meta information, 460
  use FM as APIs in ABAP/4, 460, 462
XXL_FULL_API, 460
XXL_SIMPLE_API, 462